Disaggregated Imaging Spacecraft Constellation Optimization with a Genetic Algorithm

THESIS

Evelyn A. Abbate, 2nd Lieutenant, USAF

AFIT-ENY-14-M-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT-ENY-14-M-02

Disaggregated Imaging Spacecraft Constellation Optimization with a Genetic Algorithm

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Evelyn A. Abbate, B.S., Astronautical Engineering

2nd Lieutenant, USAF

March 2014

AFIT-ENY-14-M-02

Disaggregated Imaging Spacecraft Constellation Optimization with a Genetic Algorithm

Evelyn A. Abbate, B.S., Astronautical Engineering
2nd Lieutenant, USAF

Approved:

//signed//	7 Mar 2014
J. T. Black, PhD (Chairman)	date
//signed//	7 Mar 2014
B. J. Ayres, PhD (Member)	date
//signed//	7 Mar 2014
Col. T. J. Lawrence, PhD (Member)	date

Abstract

This research is an extension of work using a genetic algorithm to optimize certain parameters of a disaggregated constellation for most cost-effective coverage. This work looks at imaging sensor coverage of a target deck. The US Army has several current projects that use disaggregated constellations with imaging payloads to provide warfighters with pertinent area data and images, such as Kestrel Eye and Nano Eye, to which this work is applicable.

Parameters varied in this optimization affect Walker constellation characteristics, orbital elements, and sensor size. Walker parameter variables are number of planes, number of satellites per plane, true anomaly spread, and RAAN increment. All classical orbital elements are variable, although a circular, low-Earth orbit is assumed. Sensor size is varied dependent upon sensor diameter. These parameters are applied to constellations of small satellites (under 500 kilograms) and large satellites (300 to 5200 kilograms).

The Unmanned Spacecraft Cost Model (USCM) and the Small Spacecraft Cost Model (SSCM) are used to roughly determine the cost of each proposed mission. The sensor effectiveness is determined by the General Imaging Quality Equation (GIQE).

The model created in this work produces notional results for the Kestrel Eye program, yielding a 15 minute response time for a given target deck, but can be customized to optimize any satellite architecture problem. This work should be referenced by sponsors to save money and increase capabilities as space architectures are made more resilient and efficient.

Acknowledgements

There are many people in my life who have helped me to get to where I am today. My mother has stood by my side since the day I was born, and Kate has been with me from day one of basic training, through the good and the bad. I also owe my thanks to my instructors back at the Academy, especially Mr. Gary Payton for providing sources and contacts for my research, and Col (ret) Jack Anthony for encouragement and being a part of my application to this program.

The instructors at AFIT are invaluable. I especially want to thank my advisor, Dr. Jonathan Black, for helping me put this project together and being available every step of the way. Finally, I want to thank Maj. Rob Thompson for taking me under his wing and allowing me to be a part of his research. I learned a lot about disaggregation, space, and the Air Force during our months working together. I wish him the best of luck as he completes his own work.

Evelyn A. Abbate

Contents

Page

List of Figures

List of Tables

List of Symbols

Symbol		Page
t_{int}	integration time	82
R_S	scene radiance	82
R_H	haze radiance	82

List of Abbreviations

Abbreviation		Page
NOAA	National Oceanic and Atmospheric Administration	39
POES	Polar-orbiting Operational Environmental Satellites	39
CGS	Common Ground System	39
SBIR	small business innovation research	41
USCM8	Unmanned Space Vehicle Cost Model, version 8	44
CER	cost estimating relationship	44
NASA	National Aeronautics and Space Administration	44
NICM	NASA Instrument Cost Model	44
SSCM	Small Spacecraft Cost Model	45
TFU	Theoretical First Unit	46
NRE	Non-Recurring Engineering	46
NIIRS	National Image Interpretability Rating Scale	48
GIQE	General Image Quality Equation	48
FOV	field of view	48
GSD	ground sample distance	80
RER	relative edge response	80
MTF	modulation transfer function	80
ER	edge response	81
SNR	signal-to-noise ratio	81
VNIR	visible near infra-red	82
IAT	integration and test	86

Disaggregated Imaging Spacecraft Constellation Optimization with a Genetic Algorithm

I. Introduction

Many of man's great accomplishments have been due to unconventional thinking. The space industry is no different; although we have made great strides thus far, it seems that the next big breakthrough will require a change in mindset. Disaggregation has lately been discussed as a revolutionary means to facilitate the industry's next leap forward in space exploration, and the Air Force is considering it as a means to enhance resiliency of space systems. The following research will assert that disaggregated architectures and fractionated systems are a resilient and cost-effective way to conduct imaging and surveillance operations, and will analyze the results of its implementation on existing concepts.

1.1 Motivation

The government typically imposes many requirements on a single mission, which is known as aggregation. Aggregation has a number of unintended but harmful consequences, and the application of disaggregation combined with mixed constellations has been endorsed by present and former commanders of Air Force Space Command (AFSPC) and US Strategic Command (STRATCOM) [1]. In MilSat Magazine, MILSATCOM director Ron Burch focuses on the U.S. Military Satellite Communications Systems Directorate (MILSATCOM), an example of a complex system that suffers from specific consequences of aggregation: "fewer assets, increased cost and schedule uncertainties, and increasing delays in the fielding of new capabilities to support the warfighter [2]." Similar results can be seen across the board. All this to say, the need for disaggregation in space has been acknowledged, and there is already much research in this area, which will be described in Chapter 2.

As more mission requirements are levied on the system and system complexity subsequently increases, timelines tend to stretch, components tend to get more expensive, and cost and schedule risks become more difficult to predict [2]. This "requirements creep" is a well-known and studied phenomenon in systems engineering, and the systems engineering process must be strictly monitored to mitigate it [3]. Disaggregation allows engineers to break the system down into components with individual requirements, and perhaps highlight areas where requirements creep is occurring.

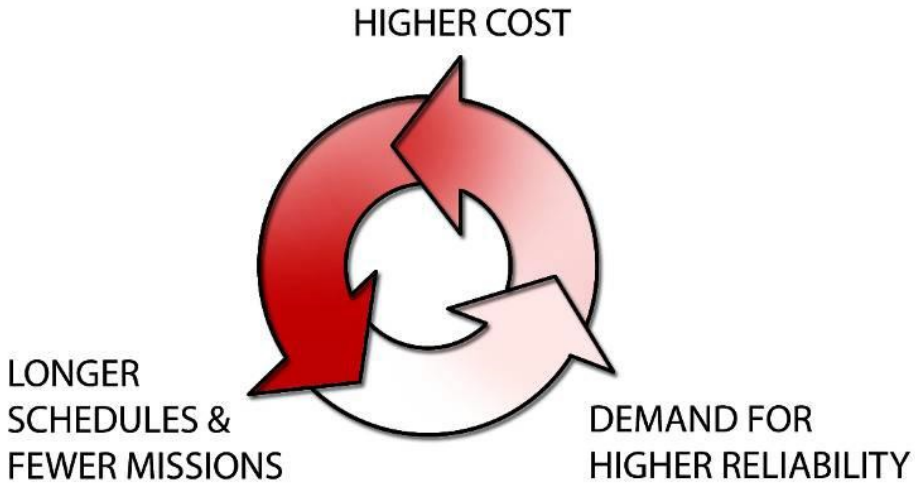


Figure 1.1: Wertz's concept of the space spiral [4]

Owen Brown and Paul Eremenko's sources of uncertainty include technology, environment, launch, demand, requirements, and funding, and cause system cost and complexities to increase, which often leads to a building spiral as uncertainties subsequently augment (Figure 1.1). One solution is to design spacecraft for these uncertainties, rather than for requirements as is the current convention [5]. By designing with uncertainties in mind, while attempting to keep cost and complexity low, a revolutionary change in space acquisitions is possible [5].

Responsiveness is also becoming more of a necessity than a novelty. "To be relevant today, systems must be responsive in hours or days, not months, years, or decades...Failing to aggressively attack this fundamental problem will mean that the US will lose its dominant position in space [4]." Major Thomas Co investigated this

shorter-term responsiveness in his dissertation, where he used electric propulsion (EP) to accomplish multiple responsive maneuvers. He concluded that EP thrusting can provide direct overflight within 1 to 2.5 days [6–8].

Launch is consistently an expensive portion of any space program, and as budgets decline, costs are escalating [9]. Brown and Emerenko cite some of the advantages and disadvantages [10]. Fractionated modules can be launched on small, responsive launch vehicles, and can reduce costs by reducing mass and volume. They also contribute to the range of launch options, as they can be launched together as a conventional payload or separately as secondary payloads [10].

Our space assets are under ever-increasing threats, both from adversaries and the space environment [9]. Warfighters depend on space systems for combat support [9], and have a need for responsive space capabilities. Disaggregated systems have the attribute of modularity, and it follows that they are flexible as pieces can be added or exchanged at any time. Adversaries reacting to Desert Storm are increasing their space capabilities [11], China is developing launch-on-demand systems, and "the Soviets/Russians have had systems in inventory and launch-on-demand (within hours) for over 30 years [4]." The space environment itself is getting more hazardous, with increased spacecraft traffic, competition over the electromagnetic spectrum, and more debris [11]. Entities must vie for a position along the crowded geosynchronous (GEO) belt [12]. The smaller size of disaggregated systems decreases the probability of collision and allows for easier atmospheric reentry. Based on this evidence, disaggregation can add resiliency to current space systems, allowing them to better survive in these perilous conditions.

Moore's Law says that technology is improving at an exponential rate, which is great for most industries. But in the space industry, it is hard to take advantage of this trend since a typical government satellite acquisition program takes many years to implement, and by the time the satellite is launched the technology is out of date. "A program that takes 10 years to design and build and then lives on-orbit

for 15 years is operating at roughly 1/4000 (0.00025) of the speed and throughout [*sic*] currently available [4]." With current standard government space acquisition, inserting new capabilities and technologies can take decades [9]. Missions must be redefined so they can be accomplished by small spacecraft [4]. The modularity of a disaggregated system would allow technology to be updated as quickly as a launch could be found.

As Space and Missile Systems Center commander Lt. Gen. Ellen Pawlikowski puts it, current systems have poor resilience and concentrated capabilities, making them "good targets that are hard to defend [9]." If just one aspect of these systems is incapacitated, mission capabilities are crippled [9]. Disaggregating these systems could allow the mission to continue as planned with only minor setbacks due to loss of a single functionality.

In light of all these challenges, disaggregation is rightly a hot topic for space systems of the future. Brown points out that there is a need to transition from large monolithic spacecraft with maximum reliability and lifetime to more innovative, agile, and economically sensible architectures [13], and AFSPC commander General William Shelton pushes for more resilience to threats in the space and cyber domains [14]. Shelton is quoted in a SpaceNews article, saying of our military space programs, "I'm a big proponent of pushing it to commercial if we can [15]." Most recently, in a 2014 published article [16], Gen. Shelton reemphasized the need for balancing "required capability, affordability, and resilience," comparing the difficulty of the process to turning the Queen Mary [16].

1.2 Problem Statement

According to the AFSPC White Paper, the potential benefits of functional disaggregation may be significant but are not yet proven [11]. This thesis is an exploration of the benefits versus the costs of disaggregated architectures. In his paper on distributed satellite system optimization, Cyrus Jilla notes that "the trade space for distributed satellite systems (DSS) can be enormous - too large to enumerate, ana-

lyze, and compare all possible system architectures [17]," and as such we are going to perform an analysis of a single case that may or may not be applicable to other cases. The research uses Matlab's genetic algorithm (GA) tool in conjunction with Analytical Graphics, Inc. (AGI) Systems Toolkit (STK) simulations to minimize the cost of a constellation of satellites with disaggregated imaging payloads. The cost will be calculated by standard cost models that take into account the number of spacecraft and sensor aperture diameter. The Matlab-STK Component Object Model (COM) interface will be utilized via a Matlab function to generate a scenario within Matlab's GA calculator. The principles of disaggregation will be applied to a simulated imaging constellation and the results analyzed to qualify the benefits of such methods on that type of system.

This thesis is an extension of Major Robert Thompson's work, which is based on calculating global coverage with a Walker constellation. This thesis focuses on a given target deck and spacecraft with targeted sensors, separating it from what Maj. Thompson has already done with push-broom and whisk-broom scanning sensors.

1.3 Definitions

Disaggregation is a broad concept with many definitions depending on the specific field of interest. Maj. Gen. Thomas Taverney refers to disaggregation in the acquisitions context as "the concept of splitting the missions into larger numbers of smaller and more consistent missions/satellites with achievable requirements...to confront and reduce overall acquisition costs, and simplify acquisition rules and strategies [1]." Space disaggregation is also defined by an Air Force Space Command (AFSPC) White Paper as "the dispersion of space-based missions, functions or sensors across multiple systems spanning one or more orbital plane, platform, host or domain [11]." In this paper, disaggregation will refer to capabilities separated onto multiple platforms (space vehicles) in a constellation or formation.

Disaggregated systems are also sometimes referred to as *distributed space systems*, or *distributed satellite systems (DSS)*. "A distributed space system is defined

as a system of multiple satellites designed to work together in a coordinated fashion to perform a mission" (Shaw, 2000 [17]). Horst's conference proceeding lists some types of DSS, such as formation flight, satellite clusters, and fractionated spacecraft; some referenced examples are TechSat21 and System F6 [18]. Another example of a disaggregated system is the Global Positioning System (GPS). Each payload cannot accomplish its mission individually, but all together the system is "a robust, affordable, and resilient architecture" that is invaluable to modern everyday life. GPS routinely integrates new technologies, which is a novel feature of disaggregated systems. The GPS 3 system pairs a generic commercial bus with a specialized payload, following a common tenet of disaggregation [9].

The two primary methods of disaggregation, according to Burch, are *decomposition* of a monolithic system into many smaller systems, and *augmentation* of an existing system by adding smaller systems [2]. The Iridium system is an example of decomposition, and the Space Based Infra-Red System (SBIRS) utilizes a form of augmentation, as highly elliptical orbiting (HEO) payloads can be seen as augmenting large GEO spacecraft [2]. Decomposition can be broken down further into *fractionation* and *functional disaggregation*, which can both utilize *multiple orbital planes*. Fractionation is decomposition of subsystems, and functional disaggregation is decomposition of the mission. Multiple orbital planes puts some spacecraft type in one orbit, at a particular altitude and inclination, and other spacecraft types at other altitudes and inclinations. Augmentation usually involves hosted payloads, or adding constellations of smaller satellites to existing constellations. All of these variations are illustrated in Figure 1.2.

Disaggregation is often used interchangeably with its subset of *fractionation*, but this is not technically correct. Brad Tousley, director of the U.S. Defense Advanced Research Projects Agency's (DARPA) Tactical Technology Office, gave some insight into these concepts in an interview with Space News in May 2013. Disaggregation, as distinguished from fractionation, is a broader concept that "entails breaking up the mission sets of large spacecraft and dispersing them among smaller satellites [19]."

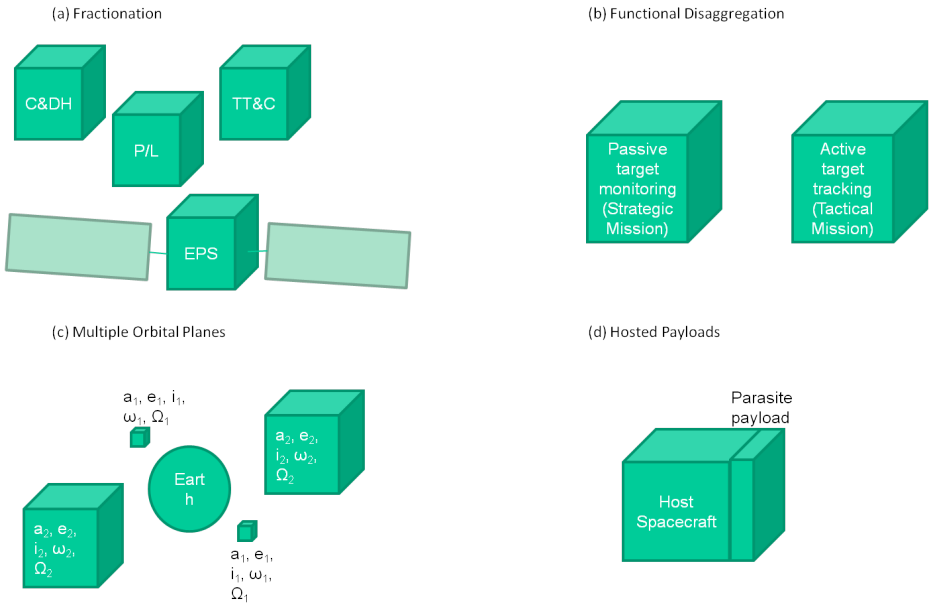


Figure 1.2: Types of disaggregation. Each cube represents an independent space vehicle, or module. In fractionation (a), each subsystem is broken off into its own vehicle, and the whole system flies in a cluster with a proximity that allows necessary data and resource sharing. In functional disaggregation (b), the mission is broken down by mission, and each independent mission has its own vehicle. With multiple orbital planes (c), a constellation of satellites with common missions are used in different orbits to maximize coverage and survivability. Multiple orbital planes can be used in conjunction with other methods of disaggregation. Hosted payloads (d) are simple missions that can be added to existing large spacecraft. Hosted payloads disaggregate one spacecraft by aggregating with another spacecraft.

Fractionation can be described further as "dispersing the functions of a single satellite across several smaller platforms [19]," or "multiple subcomponents interact[ing] on orbit to holistically create the capability of a single monolithic satellite [11]." Brown and Eremenko define fractionation as "the decomposition of a system into distinct modules which, once 'assembled' on orbit, deliver the capability of the original monolithic system [5]." Jerry Sellers includes a comic in *Understanding Space* showing what a spacecraft would look like if it was designed by just one type of engineer (Figure 1.3). This is remarkably representative of the concept of fractionated spacecraft.

Brown and Emerenko further break down the concept of fractionation into two categories: homogeneous fractionation, in which the modules are identical or functionally similar, and heterogeneous fractionation, where each element is unique and

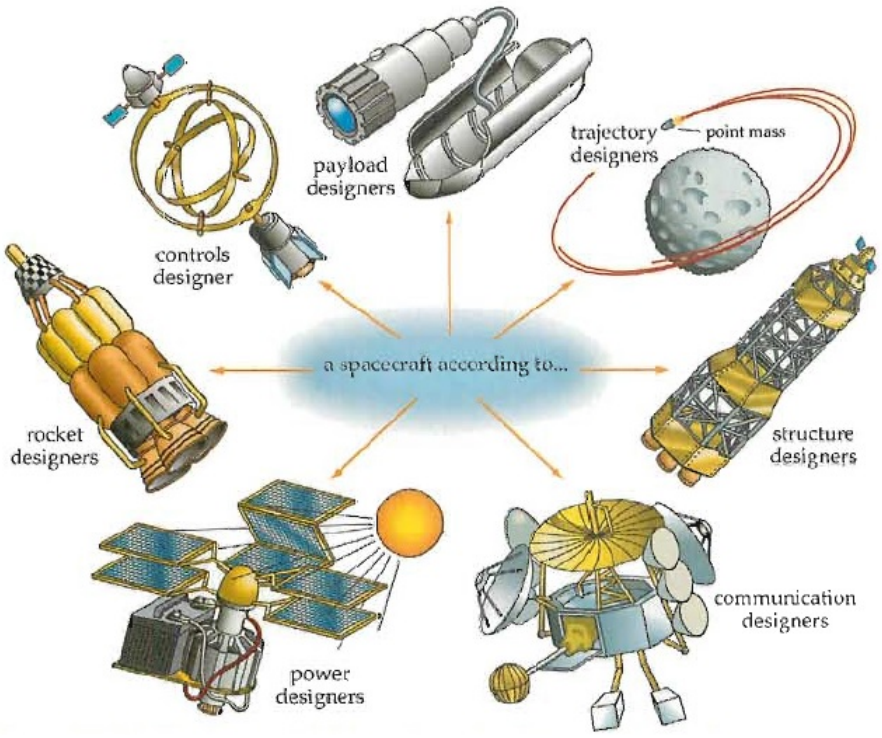


Figure 1.3: Jerry Sellers' depiction of a spacecraft as designed by a single type of engineer [20]. This is a useful illustration when talking about fractionation.

modules may interact wirelessly [10]. According to Charlotte Mathieu, "spacecraft fractionation transforms a large monolithic spacecraft into smaller modules," with homogeneous fractionation referring to a cluster of identical spacecraft. She says that "heterogeneous fractionation divides the spacecraft into its functional elements," implying that reusable bus modules could be able to support various payloads [21].

Functional disaggregation, or breaking down the mission, is a slightly newer concept. An AFSPC White Paper defines functional disaggregation as "the dispersion of sensors or distinct sub-missions onto separate platforms that were previously hosted on a single system [11]." The mission is broken down into strategic and tactical components, with each piece being accomplished by an individual module or space vehicle. As an example, visualize a target tracking system, which might have a strategic mission of global passive coverage, and a tactical mission of active target tracking in more concentrated regions.

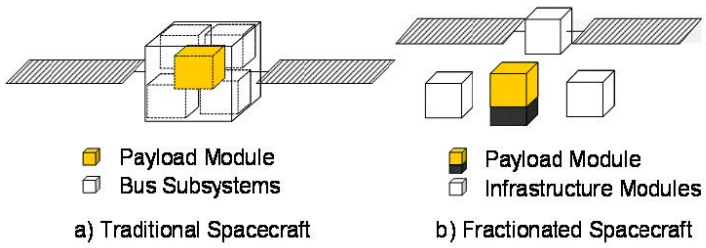


Figure 1.4: Charlotte Mathieu's illustration of the difference between traditional and fractionated spacecraft. "Spacecraft fractionation transforms a large monolithic spacecraft into smaller modules [22]."

Fractionated and functionally disaggregated systems could both utilize *multiple orbital planes*, which is especially useful in heterogeneous systems. If an imaging system contains two types of space vehicles, distinguished by their sizes, the smaller spacecraft can compensate for a smaller instrument aperture diameter with a lower altitude orbit, while the larger aperture diameter can handle the resolution decrease that results from the higher altitude. The space vehicles in the constellation all have the same mission, but the mixed constellation yields better survivability and coverage.

Finally, *hosted payloads* add a mission to an already-existing commercial or government spacecraft. This capability can make a complex system simpler by detaching one component and appending it to an existing spacecraft, making use of the host spacecraft's orbital characteristics and subsystems.

The orbital imaging system studied in this research makes use of heterogeneous fractionation over multiple orbital planes, as will be seen in Chapter 3. A constellation was found to accomplish the mission in under two minutes for a Middle East target deck, or more reasonably within 30 minutes using an Ohio target deck. Using the approach developed in this thesis, an actual target deck can be entered into the algorithm to design an optimal constellation.

1.4 Thesis Overview

Chapter II provides background research on the pros and cons of disaggregation, pros and cons of various optimizers, and an overview of the STK COM interface, as

well as a description of the Army's space-based responsive imaging projects. Chapter III is a detailed description of the methodology used in this research. Chapter IV provides the results of the simulation and research, and Chapter V presents conclusions and suggests future work on this topic.

II. Background

This chapter presents a summary of topics relevant to disaggregation research and optimization methods. It covers the various types of disaggregation and expounds upon its uses. Pros and cons of various optimizers are explored, and the use of the STK-Matlab COM interface is discussed. Finally, some applications are identified and discussed.

2.1 Concepts of Disaggregation

To fully understand disaggregation, there are some commonly-used terms that need to be described.

Disaggregation makes use of many small spacecraft in *cluster formations*. These cluster formations are not meant to be an organized formation; it simply means that they are close enough to communicate with each other. This concept is contrasted with the concept of the *monolithic spacecraft*, from which systems are disaggregated. A monolithic spacecraft is one which does not have readily-separable modules, and can be treated as a single, independent system.

Another concept intrinsic to disaggregation is *modularity*. A modular system is a system of subsystems, where each spacecraft in the fractionated space architecture can be thought of as a subsystem [21]. Although there are several definitions of modularity, the term "modular" in this paper will refer to an architecture or constellation made of independent, unique, and interchangeable spacecraft (modules) working together toward a common objective; this paper does not discuss interchangeable subsystem modules contained within one spacecraft.

Finally, Brown and Eremenko define a couple more uncommon terms in relation to disaggregation. They use the term "uncertainty" in place of risk and opportunity, where risk refers to situations leading to system degradation, and opportunity describes potential improvement in the system [5]. Larson et al. list the sources of uncertainty as statistics, input variables, and "unknown unknowns," or unpredictable

events. Possible responses to uncertainty are: avoid, mitigate, transfer, or accept risk; exploit, enhance, share, or accept opportunity [23].

Brown and Eremenko also use the term responsive space, which is "the capability of space systems to respond rapidly to uncertainty [5]." In this paper, responsive systems have the capability to be launched or changed in a timely manner based on user needs and requests.

James Wertz expands on the concept of responsive space in Chapter 13 of the New SMAD using his coined concept of "reinventing space:"

> Reinventing space is using modern technology and old-fashioned drive, determination, and some willingness to accept risk to do much more, much faster, with fewer resources [4].

Wertz emphasizes the importance of mission diversity, or mixing large, expensive programs with small, quick, flexible systems. The goal is to accomplish the same mission objectives without the expenses of traditional programs. "If we can do some entire missions, and parts of other missions, far quicker and at much lower cost, we can expand what we can do in space, be more responsive to the needs of the end user, and do more with fewer resources [4]."

The goal of disaggregated architectures is to improve resiliency while reducing costs, increasing modularity, and simplifying complex systems. If this is not possible, we will need to find a different strategy to bring about the future of space architecture.

In addition to disaggregation terms, a basic understanding of astrodynamics is helpful in following this analysis. There are six parameters necessary to describe the orbit of a spacecraft around Earth, known as classical orbital elements (COE). These consist of semi-major axis (a), eccentricity (ecc), inclination (incl), right ascension of the ascending node (RAAN), argument of perigee (ω), and true anomaly (ν). A full explanation of these terms can be found in Appendix A.

Walker constellations are used in this simulation; these constellations were developed by J.G. Walker at the British Royal Aircraft Establishment to provide con-

tinuous Earth coverage with a minimal number of satellites. Walker's work concluded that a minimum of five satellites are necessary for persistent coverage [4]. Walker constellations are defined by the number of planes, the number of satellites per plane, and an interplane spacing or true anomaly phasing value [24]. All satellites in the Walker constellation have the same altitude and inclination as the seed satellite, and other orbital parameters are defined by the specified Walker parameters. Examples of existing Walker constellations are GPS and Iridium.

2.2 Disaggregated Systems

There has been much buzz about disaggregation of space systems as of late. A paradigm shift to these disaggregated systems seems to be in order. An AFSPC White Paper claims that US reliance on space was a glaring vulnerability after *Desert Storm*, leading to international progress in space tactics. Not to mention increased debris in the space environment, and fiscal challenges exacerbating already steep space system costs [11]. The industry would benefit financially and technologically from a transition to disaggregated systems, although this is not to say that such a radical change would be easy or fast.

However, the excitement should be moderated; not all systems are guaranteed to benefit in the same ways from disaggregation. One trade-off is similarity of components [2]: would the system benefit more from redundancy (identical components) or dispersed capability (unique components), or something in between? Engineers must also decide whether it is more cost-effective to forego disaggregation altogether and instead revert to the traditional approach to space system design; that decision may depend on system requirements as well as the number of stake holders. If the system has only a few simple requirements and is not likely to change due to the demands of various competing entities, the benefits of disaggregation could be outweighed by the labor and research needed to implement such strategies. Former National Security Council space policy director Peter Marquez is paraphrased in Laura Delgado's Space Policy Online article: "The goal ought to be, he said, not to adopt disaggregation in

all systems, since 'large systems have their place,' but instead to seek greater diversification." In the same article, Boeing Director of Missions and Programs of Commercial Satellite Services William P. (Bill) Reiner is said to support "a mix of approaches, with the end goal to find ways to be 'more responsive, resilient and cost-effective.'" Again in the same article, Lockheed Martin vice president of strategic planning Marc Berkowitz is paraphrased, "the community should not rule out big systems, but adopt whatever size best addresses the specific mission. ...the assertion that disaggregation will lead to cost savings should be validated [25]." As will be discussed, System F6 was a project designed to look into the viability of disaggregation, but was cancelled because it did not have a sustainable mission [19]. Instead of designing missions to study disaggregation, disaggregation should be applied to existing mission needs, and the results analyzed to determine the benefits that could be realized.

2.2.1 Previous and Current Work. A start to disaggregation is modularity. Competition has driven the commercial space market to "maximize the use of common bus components and modular structures, providing a core capability that enables them to configure, rather than redesign, a satellite to meet its specific mission requirements." Many modular designs exist and are used to facilitate production; even the Air Force Institute of Technology's (AFIT) space vehicle design course utilizes modular bus components so that students can design and build a payload to function with those bus components in only 30 weeks.

Modularity is only one asset of fractionated systems. According to AFSPC, the advantages to fractionation are single subcomponent replacement, reduced payload mass, lower integration costs, and higher tolerance of risk [11]. DARPA began investigating fractionation concepts with their Future, Fast, Flexible, Fractionated Free-flying Spacecraft United by Information Exchange (System F6) program, which was officially cancelled in May 2013 [19]. A rendition of this concept is shown in Figure 2.1. System F6 was supposed to demonstrate the benefits of disaggregation; separated spacecraft were meant to exchange data. Brad Tousley, director of DARPA's Tactical

Technology Office, emphasized the cancellation did not imply a lack of interest into the broader concept of disaggregation; he believes the program would have faired better with a more distinct mission rather than serving as a technology demonstration [19].

Figure 2.1: DARPA's System F6, which was supposed to demonstrate the benefits of disaggregation. It was cancelled in May 2013 [22].

DARPA is working on other programs that utilize disaggregation, such as Phoenix and SeeMe. Phoenix is supposed to demonstrate life-extension of GEO satellites by replacing spent components or fuel, while SeeMe is similar to the Army's many concepts of responsive imaging spacecraft that can deliver data to handheld devices in the field [19]. According to a recent Space News article, SeeMe was recommended for termination in the 2014 defense spending bill [26]. The fiscal environment has become increasingly more restrictive over the past few years, and programs must be extremely vital to survive to launch and on-orbit operations. The goal of this research is to establish a method for efficient constellation design in order to strengthen future project proposals and prevent cancellation.

Mass	20 kg (dry) / 80 kg (wet)
Minimum working elevation angle	20 degrees
Minimum field of view at nadir	0.8 km x 0.8 km
Swath width at min / max altitude	500 km / 2000 km
Orbital inclination	3 to 5 degrees higher than latitude of the target

Table 2.1: NanoEye Specifications [29]

Cost	$1M per spacecraft in production mode
Mass	18 kg (dry) / 25 kg (wet)
Altitude	500 km
Focal length	10 m
Field of view	5.8 x 3.8 km
GSD	1.5 m at nadir
Downlink rate	1 Mbs
Tasking-to-product cycle	10 minutes

Table 2.2: Kestrel Eye Specifications [30–32]

The Army's imaging programs include Kestrel Eye and NanoEye. These programs are solutions for providing the warfighter real-time imagery of areas of operation [27]. The specifications for NanoEye are taken from Captain Steven Ingraham's thesis and are shown in Table 2.1. NanoEye's fact sheet from the US Army Space and Missile Defense Command/Army Forces Strategic Command claims that "NanoEye is a low-cost microsatellite providing responsive, tactical imaging from Low Earth Orbit to the ground component Warfighter [28]." The program emphasizes low cost, and aims to provide desired images within minutes to the ground forces that are directly tasking the satellites. NanoEye will use on-board propulsion to get better resolution images from lower altitudes [28].

Kestrel Eye is another "small, low cost, visible imagery satellite demonstrator that offers the tactical-level ground component Warfighter real-time imagery [30]." Like NanoEye, Kestrel Eye will be directly tasked by the ground forces, will provide desired imagery within about 10 minutes, and is meant to extend the Unmanned Aerial Vehicle paradigm into space. Kestrel Eye aims to be small in size but great in number, and cost around $1 million per spacecraft in production mode [30]. Kestrel Eye's specifications are listed in Table 2.2 on page 16 and are used as a reference for this research. A basic concept of operations for these programs can be found in

Chapter 3, as this research attempts to recreate and analyze a similar concept.

The U.S. Air Force is currently looking at revamping GPS plans. Warren Ferster's SpaceNews article details some of the nine options under examination, from a constellation of small satellites called NavSats or augmentation by geostationary satellites to continuing with the expensive but low-risk Lockheed Martin GPS 3 constellation [33]. The NavSats may sound like a good deal, but involve a high level of cost, performance, reliability, and lifetime risk due to low levels of technological maturity. "The Air Force late last year awarded contracts...to three companies - Boeing, ITT Exelis and Surrey Satellite Technology U.S. - to study low-cost satellites to augment the GPS constellation [33]." This thesis furthers the Air Force's intentions by studying cost-effective constellations that can be used to augment existing satellites and constellations as needs develop and change.

The European Space Agency is working on FUEGOSAT, a 12-satellite LEO fire observation mission. The mission involves three satellites, each on a different orbital plane. The FUEGOSAT payload is a multi-spectral push-broom radiometer [34]. The conclusion of this thesis will be pertinent to such missions, as the number of satellites needed in the constellation, and so the overall mission cost, may be reduced.

Here at AFIT, Major Robert Thompson has taken interest in disaggregation. Starting from a model in Excel, he created the original versions of the Sensor_performance.m and cost.m Matlab code (see Appendix D). By adapting those pieces of code, this thesis will be an extension of his work, which he will continue as he completes his doctorate.

A couple of theses from AFIT are also related to this topic. A thesis on Air Force Research Laboratory's (AFRL) TACSAT-2 investigates the utility of matched inclination orbits for tactical users and logistics impacts for the Iraqi theater [35]. The specifications for TACSAT are listed in Table 2.3. This thesis takes their investigation a step further by looking at varying all orbital parameters for a constellation

Altitude	350 km
Eccentricity	0
Inclination	Latitude of target
Mass	352 kg
Slew rate	0.5 deg/sec
Target	Iraqi theater

Table 2.3: TACSAT Specifications [35]

monitoring the Iraqi theater. The TACSAT paper found a revisit time of 52 minutes with 10 satellites and a revisit time of 22 minutes for 20 satellites.

Captain Steven Ingraham's thesis uses maneuvering satellites to cover Dayton, OH [29], whereas this thesis assumes the satellites are in constant orbits and are incapable of tactically maneuvering. But Capt. Ingraham makes the general assertion that for maximum coverage, the inclination should be the sum of the target latitude and the Earth central angle of the spacecraft. His equations are:

$$\sin\rho = \frac{R_E}{R_E + h}$$

$$\sin\eta_{max} = \sin\rho\cos\epsilon_{min}$$

$$\lambda_{max} = \frac{\pi}{2} - \epsilon_{min} - \eta_{max}$$

$$Incl_{optimal} = \lambda_{max} + \lambda_{target} \tag{2.1}$$

$$\tag{2.2}$$

where R_E is the radius of the Earth (6378 km), h is the spacecraft altitude, ρ is the angle from the spacecraft to the local horizon, η is the limits within minimum elevation angle, ϵ is the minimum elevation angle, and λ_{max} is the Earth central angle [29], as illustrated in Figure 2.2 on page 19. λ_{target} is the latitude of the target. This model will be analyzed further in Chapter 4.

2.2.2 Benefits of Disaggregation. Burch states that new systems should aim to be less complex than previously designed systems, and should allow for system growth [2]. Disaggregated systems accomplish this goal; they take less development

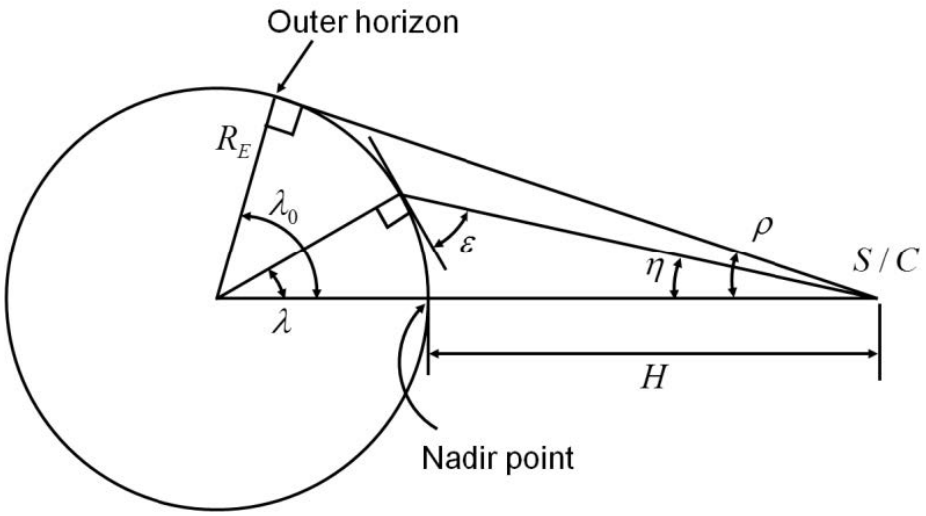


Figure 2.2: Spacecraft angular geometry for optimal inclination calculation [29]

time, have lower cost, and are more flexible than their monolithic counterparts. Disaggregation can mean improvements in schedule, cost, and performance of a space program.

2.2.2.1 Schedule. Modular structures allow engineers to "configure, rather than redesign, a satellite to meet its specific mission requirements [9]." Reusing existing modules and not starting from scratch can significantly reduce the temporal length of a program. And since fractionated system modules are not highly interconnected, subsystems can be developed in parallel, further saving production time on the overall system [21]. A system developed in this way is more flexible, and could adapt to fluid requirements.

Improved schedule not only means faster production of capabilities, but it also allows a refresh of on-orbit technologies that often become obsolete before they are launched [11]. Since "only what is needed is developed, manufactured, and launched [36]," space systems are more efficient and can concentrate on meeting specific requirements.

2.2.2.2 Cost. A common reason for cost increase is the difficulty of integrating multiple payloads onto a single bus, which additionally can cause schedule challenges [11]. Integration is a difficult part of space system design, especially for more complex systems; there are many physical and non-physical interfaces that must be worked through, and getting each component to talk to the others is a challenge. Additionally, all the necessary components must fit into as small a volume as possible. Brown and Eremenko point out that interconnected modules with only a wireless interface are much simpler to integrate than a large spacecraft with technical and programmatic connections, which includes physical interactions and interdependence regarding readiness [10].

An important element in system cost can be attitude control requirements. Some components have stricter attitude control requirements while others have almost no attitude control requirements. Attitude determination and control (ADC) systems can be significantly downsized by breaking up the spacecraft [10]. Each module could have its own ADC system, with only as much accuracy as necessary. Some modules may not even need to include ADC at all.

Lifetime costs are another cost driver of any space system. The technology obsolescence of monolithic spacecraft increases recurring costs over time, as well as production costs as the system must be redesigned to incorporate new components [2]. Redesign may also be necessary as some components expire faster than others, and radiation can shorten the on-orbit lifetime of spacecraft operating in more hostile orbital environments. "Hardware with similar design lifetimes can be aggregated on modules thus creating some short- and some long-lived modules [10]." Space systems would no longer be limited to the lifetime of their shortest-lived component. This idea of grouping by lifetime may also lead to a precedent for disaggregated system design.

Security is an important consideration, especially on intelligence payloads. If only one module has strict security requirements, the other payload modules can be

developed for a much lower price [10]. Security is another aspect to consider when splitting up capabilities.

Launching a mission using multiple launch opportunities leads to fewer launch constraints and less financial risk [21]. Multiple launches can be thought of as a kind of insurance, which is a valuable commodity in the space industry. In fact, in a cost-benefit analysis of fractionated architectures, Brown et al. points out that "reducing cost risk by architectural means, such as fractionation, is effectively equivalent to self-insuring the system during its launch and operations phases." Figure 2.3 is taken from the Department of Transportation by Brown, and shows industry-wide insurance rates for launch plus one year of on-orbit operations (blue) and annual operations (red) costs [13]. Although more launches would be inherently more expensive, using separate launch vehicles distributes risk and reduces chances of total mission failure, and makes the fruits of the financial investment more valuable.

Burch also alludes to the ever-important aspect of cost certainty [2]. Once the cost is calculated, that is not the end; unexpected circumstances often arise and push the price even higher. "As program timelines lengthen, these fixed costs increase proportionally," especially with bigger and more complex systems [2]. The old adage, "time is money," holds very true in the space industry. It is prudent to use disaggregation as a strategy to decrease the time of production, and therefore increase the cost certainty of the project.

The caveat to these ideas is that the greater number of launches required imposes some complications. For systems such as MILSATCOM, as discussed by Burch, U.S. space policy restricts potential launch vehicles to those supplied by the U.S. [2]. However, the growth of U.S. commercial space launch suppliers like SpaceX will conceivably mitigate this issue.

Brown's research shows that "the overall cost-benefit proposition may close in the fractionated case and fail to close in the monolithic case [13]." The actual results can be seen in Figure 2.4 on page 23. This means that, in the long run,

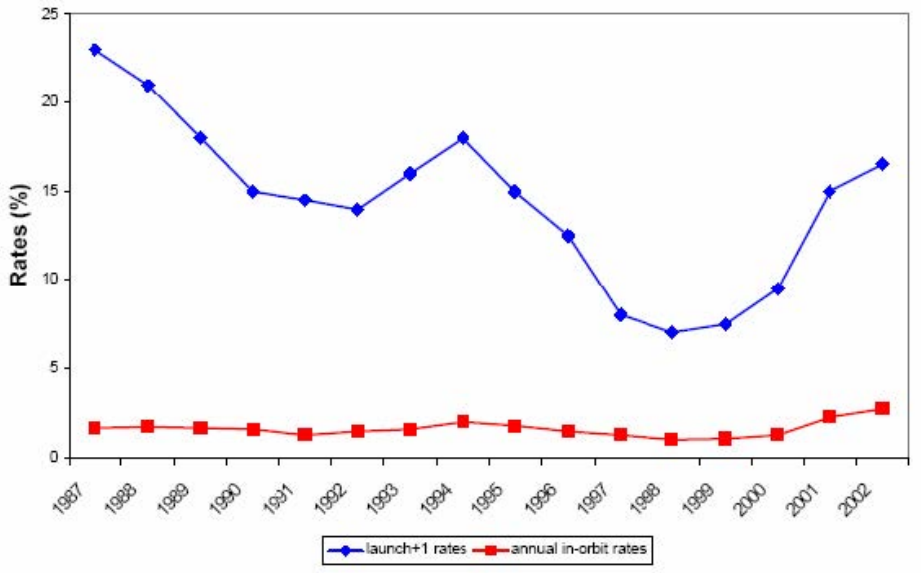
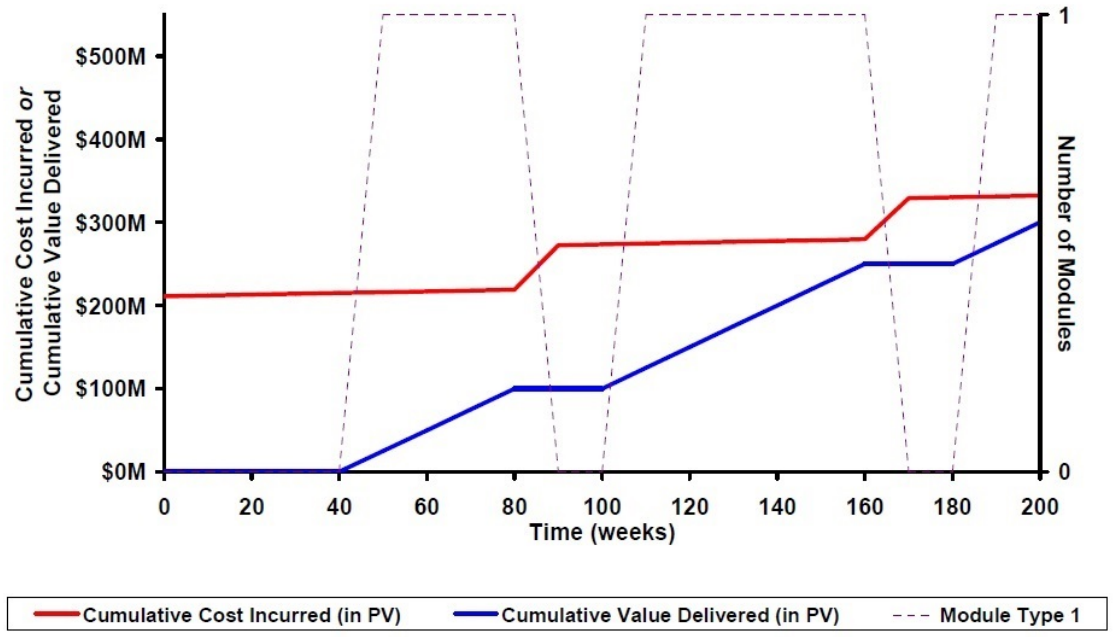


Figure 2.3: Average Launch Insurance Rates [37]

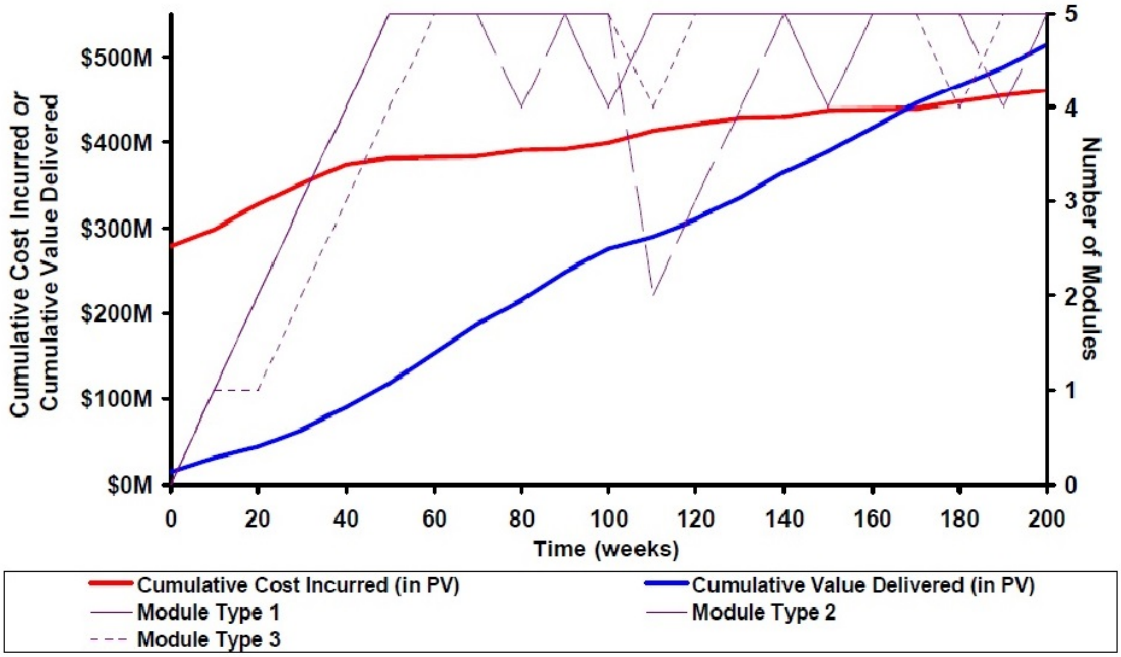
fractionation turns out to be more cost effective. Pawlikowski notes that "the cost of higher quantities are offset by savings from hosting, continuous production lines, commercial bus procurements, smaller and less-complex satellites, more-frequent and lower-cost launch, and a more tailored approach to mission assurance [9]."

2.2.2.3 Performance. This section will show that modular architectures are more flexible, responsive, and survivable than traditional monolithic spacecraft [21], against both hostile adversaries and environmental threats [11].

Disaggregated architectures are flexible, decreasing program risk. In her Space Policy Online article, Delgado states that Bill Reiner "emphasized disaggregation as an opportunity for more flexibility as military missions change [25]." A modular system, as discussed before, allows modules to be developed with different lifetimes [21]. These systems are less complex, which may encourage sponsors to accept more risk [11], allowing missions more capability and overall value. In fact, Burch [2] references the exponential relationship between costs and complexity, and asserts that disaggregated systems are able to remain below the knee of the complexity curve. Disaggregated systems present stakeholders with a wider array of options throughout



(a) Monolithic Spacecraft Cost-benefit Analysis



(b) Fractionated Architecture Cost-benefit Analysis

Figure 2.4: Monolithic Spacecraft versus Fractionated Architecture Cost-benefit Analysis [13]

the system lifetime, which in essence is the definition of flexibility [10]. Disaggregated systems are also more adaptable because they are inherently redundant; a capability lost on one module might be replaced by the same capability on another module [10]. Wertz provides an example of the flexibility provided by disaggregated systems:

> If we are launching 5 to 10 low cost observation satellites per year, then there is relatively little risk associated with introducing a new sensor technology into the satellite stream. If, for any reason, it proves unsuccessful, there are still a number of on-orbit satellites using the prior sensor technology that are still providing data [4].

Disaggregated systems are responsive. Xin Liu et al. notes that complex systems are riddled with uncertainties that are difficult to control and measure [38]. The two main uncertainties cited are user requests and weather. User needs are unpredictable and may arise unexpectedly, and weather is inherently unpredictable [38]; even with modern weather prediction technology forecasts are often wrong. Therefore, responsive systems are necessary to be flexible in this unstable kind of environment. As demands change, disaggregated systems can be incrementally upgraded without loss of capability in the meanwhile and without long wait times [13]. Systems are made ever more responsive as there are multiple launch options. Modules can be launched in pairs or groups on conventional launch vehicles, launched as secondary payloads, or launched individually on responsive vehicles such as those provided by SpaceX [10]. Whereas monolithic systems are put into place and subject to fluctuating threats, disaggregated systems are able to be changed in a much shorter amount of time in response to vulnerabilities [2].

Disaggregated systems are survivable. Capabilities lost by a single spacecraft take years to replace, at a high cost, while the loss of one of many small spacecraft may not be significantly detrimental to the mission; the small spacecraft is relatively easy to replace in a short time and the replacement may even have greater capability than the lost asset [4]. As spelled out by AFSPC, "the goal is to make attack against our systems as difficult as possible, while increasing the possibility of capability survival in the face of hostile action [11]." Disaggregated architectures have a target-spreading

characteristic, or "spatial distribution of components over a larger volume [10]." By virtue of this unique attribute, adversaries can less easily determine where and when to strike, and are less certain of success in their attack [11]. Autonomous tracking systems may also be baffled by autonomously changing geometries that disaggregated systems make possible [10]. The potential impact on the warfighter is mitigated and the impact of a satellite lost to mishap or hostile action is reduced by transitioning from a single satellite asset to more platforms and dispersed capabilities [9]. The reduced complexity of these systems significantly decreases the chances of experiencing catastrophic failures such as those had by Apollo, Challenger, Chernobyl, and Columbia [10]. Wertz argues this as well:

> If critical US military assets are destroyed...they cannot be replaced...they also become inviting targets... However, if some subset of their capability can be replaced within hours or days by inexpensive smallsats, they become a less valuable target and, because of low cost and rapid replacement, the back-up smallsats are also uninvinting targets [4].

2.2.2.4 Industry. During the design of the space Shuttle, "there was no continuous production process that would provide for changes, upgrades, maintenance, and incorporation of new technology [4]," which is something that disaggregated architectures can provide. Because of the post-launch opportunities for changes, disaggregation provides constant adaptability and allows industry to remain on the cutting edge of technology [11]. The newest equipment can be inserted onto older mission platforms, and experimented with or utilized immediately. Technology development and demonstration can be accomplished in parallel with ongoing operational programs [2], allowing more flexibility in technology advancement.

Disaggregation also encourages healthy competition, as payloads can be provided by different contractor teams. In this way, workloads are also distributed [11], leading to faster production and more highly-capable systems.

2.2.3 Challenges to Disaggregation. The biggest challenge to disaggregation may be integration, especially with multiple contractors having to interface their

systems with each other [11]. During integration, engineers must attempt to identify and solve unforeseen problems brought about by "the multitude of unknowns, uncertainties, and ill-functioning parts," and unforeseen consequences stemming from assumptions made in the design phase [23]. Although integration of disaggregated systems is easier than that of monolithic systems in a physical sense, issues need to be worked in the interfaces between modules. As modules are separated by distance in space, a wireless link must be established to enable communications between all subsystems. And with different contractors working on different pieces, coordination is essential to successful integration of the entire disaggregated system. And, of course, unforeseeable problems may arise as a result of emergence. Emergence refers to unintended consequences, good or bad, that arise during the integration of a system. Burch suggests a centralized mission planning element and new or expanded planning tools to keep track of the capabilities of each separated asset of the system [2]. Currently, interface standards are critical, and will become even more so with the advancement of disaggregation.

Another challenge is the increase in orbital debris. Using more satellites will create more scattered pieces of debris when the mission goes out of operation. Measures will need to be taken to ensure the system components are able to comply with the 25-year deorbit regulation practiced by NASA and the Department of Defense under the US National Space Policy; each module will need to either re-enter and burn up or be propelled out to a graveyard orbit. A recent debris-control report states that out of 38 low-orbiting satellites, 34 will not reenter within the required 25 years [39]. Thought may need to be put into the amount of spacecraft that are in the GEO graveyard orbit, and how to avoid overcrowding down the line. However, Wertz counter-argues this point, citing the number of tracked objects in LEO: 900 spacecraft in an environment containing 500,000 debris particles greater than one centimeter. Small spacecraft also have a small cross-section, resulting in a low probability of collision. LEO is "self-cleaning," since debris succumbs to drag and reenters

relatively quickly [4]. There is also a lot of current research in propulsion that may help address this issue.

Along with debris, there will also be greater use of the communications sprectra, increasing potential interference. However, this may also carry some benefits, as spatial isolation in the GEO belt "will inherently provide the opportunity for greater frequency reuse," and single-frequency systems are easier to plan and operate than multi-band systems [2].

Finally, Wertz writes that funding these missions may pose a challenge, since there are fewer job opportunities created by smallsat missions. Funding depends on public relations, and "small successes are not as newsworthy as expensive failures [4]." And despite savings in other areas and better insurance against risk, the raw cost of more launches is greater.

2.2.4 Unknowns. The most difficult part of any program is the "unknown unknowns." These are unpredictable events that cannot be accounted for, and do not show up until it is too late. In cost-risk analysis, using a confidence level higher than 50 percent is "prudent because analysis, at best, still excludes unknown unknowns [23]." There are a few aspects of disaggregation that have unknown or difficult to predict effects. These effects could result in either difficulty or improvement in both production and operations.

2.2.4.1 Production. Affordability was brought about in the automotive and computer industries by commoditization [10]. Similarly, disaggregation could make space more affordable. Widely-implemented fractionation may conceivably lead to assembly-line production of common bus components. These "cookie-cutter" bus modules could be used by multiple payloads with similar requirements.

The government's top-down approach to design drives unique requirements that demand a customized bus [9]. These specialized programs will need to adapt to work with less customizable, but more capable, standard bus components.

2.2.4.2 Operations. The reliability of smallsats decreases with respect to their monolithic counterparts because of cheaper production methods and greater risk. They are generally less reliable because they have less redundancy and require less rigorous testing. However, smallsats are more reliable than their monolithic counterparts in terms of simpler designs and more personal responsibility taken by engineers [4]. A key to disaggregation will be to get customers to accept less reliability in exchange for more versatility.

The complexity of many modules flying in a cluster formation may dictate more autonomy in fractionated systems [10], which could be both a blessing and a curse. Autonomous systems require fewer personnel to operate, are arguably more efficient, and are overall more convenient as less resources must be expended in the long run. However, they require more effort in the design phases. Autonomy may or may not be worth the extra effort for a space program lasting only a year or two.

Another change in operations would occur if ground segments were thought of as simply another node in the system [10]. The whole paradigm of space systems would change by considering all mission components to be modules of a single colossal disaggregated architecture. This would bring about better continuity in the system, but also requires more pieces to be integrated, and exacerbates the consequences of emergence. Attempting this would require acute attention to detail throughout the decomposition and integration processes.

Burch mentions that modern networks, such as the Internet, use distributed architectures with a high number of nodes. "Network utility increases...proportional to the number of nodes [2]," meaning that a desired trait of future highly developed distributed systems will be high number of nodes.

As a final note on unknowns, Wertz states that "because we will likely never build both a low-cost smallsat and a traditional satellite with the identical mission objectives, there is no way to prove conclusively that cost reduction is real [4]."

2.2.5 Using Disaggregation. Disaggregation has many potential benefits, and only a few foreseeable shortcomings. The challenges presented by disaggregation are manageable, though the biggest challenge seems to be starting a movement in the space industry toward its implementation including increasing the general tolerance of risk and lower reliability. It is worth looking into ways to separate spacecraft into modules, such as by life expectancies or classification requirements. A summary of the characteristics of disaggregated spacecraft versus monolithic spacecraft, from Charlotte Mathieu, is shown in Figure 2.5.

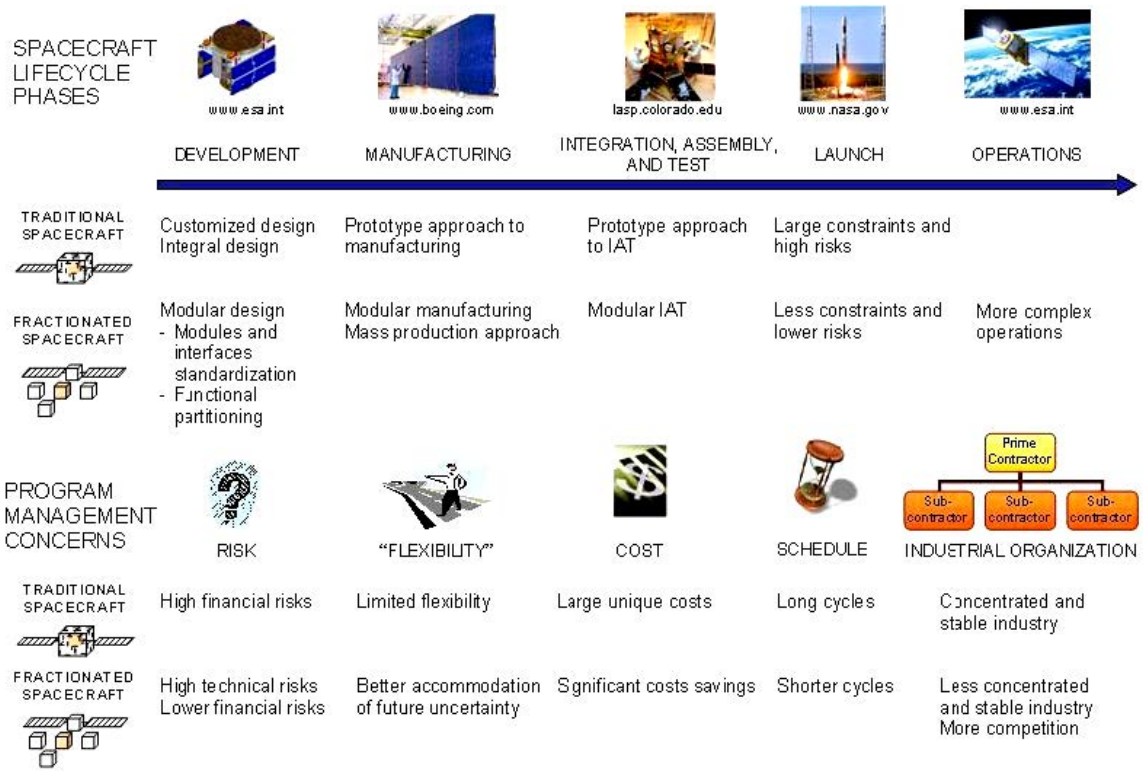


Figure 2.5: Traditional versus Fractionated Spacecraft Paradigm [40]

Establishing a method to allocate mission capabilities is a daunting task. Each component will have its own requirements, must remain unique relative to other components, and must interface with the full system [2]. Josh Hartman, CEO of the Horizons Strategy Group and principal of the Center for Strategic Space Studies (CS3), is cited in a Space Policy Online article, saying that we should examine each mission area to determine how best to disaggregate the system [25]. Hartman advo-

cates taking small steps toward disaggregation, starting with complementing existing architectures. He also hopes to see not just studies and research, but "tangible results that will help the community understand how best to adopt disaggregation to meet evolving needs [25]."

Convincing customers to accept the risk of lower reliability is another problem. The benefits wrought from disaggregated systems in cost savings and efficiency must clearly outweigh the potential losses. The best way to accomplish this is to design a program that demonstrates the merits of disaggregation and that is simple enough to not fail. Over time, more complex disaggregated systems can be attempted, as engineers grow accustomed to disaggregation concepts.

Wertz agrees that care must be taken in determining whether disaggregation is appropriate to a program. There are some tasks that simply cannot be accomplished by smallsats, and depending on the mission the cost of research and development of a smallsat system might exceed the cost of simply using a traditional architecture to accomplish objectives. Mission diversity is important in the space industry; "having the right mix of large and small satellite programs is a major step toward becoming both more efficient and more capable [4]."

2.3 Optimizers

The form of a typical optimization problem is applicable to all problems, regardless of complexity.

Minimize a cost function:

$$f(\boldsymbol{x}) = f(x_1, x_2, ..., x_n)$$

Subject to the p equality constraints:

$$h_j(\mathbf{x}) = h_j(x_1, x_2, ..., x_n) = 0; j = 1 \text{ to } p$$

and the m inequality constraints:

$$g_i(\mathbf{x}) = g_i(x_1, x_2, ..., x_n) \leq 0; i = 1 \text{ to } m \tag{2.3}$$

The solution is a vector of design variables, $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, that meets all the criteria [41]. As optimization problems become more complex, the complexity of the cost function and constraint equations increase, but the basic form remains the same.

Our problem in this research is to minimize the cost of a distributed constellation of satellites. Of course, a decision must be made regarding what type of optimization routine to use that best fits our problem needs. According to Jilla, "distributed satellite system conceptual design problems are combinatorial nonlinear problems with nonconvex trade spaces. ...as a result of the problem structure, heuristic techniques work best on DSS conceptual design optimization problems [17]." Single and multi-objective optimization methods both have their uses [17], but for now in our problem we simplify all criteria down to one objective: target coverage.

Still, Liu et al. points out that with current approaches to optimization of complex systems, the focus is on design alternatives rather than the system as a whole, resulting in a less-than-optimal outcome. Keeping that in mind, it is necessary to determine an approach that optimizes the whole system, remembering to include all the significant pieces in the model [38]. Future work will include adding complexity to make the model developed in this research more representative of actual systems.

Another issue with complex system optimization that includes many parameters is that the number of possible combinations of design parameters is nearly infinite. Time and money limit the number of combinations we can try in a model, and so it is useful to narrow down the experiment to typical parameters using such methods as design of experiments (DOE) [38].

Liu et al.'s journal article asserts that a combination of genetic algorithm (GA) for global search and improved general pattern search (IGPS) for local search would provide a high-quality, high-efficiency method for optimization of an earth observation satellite system, since "IGPS can help GA getting [sic] out of local optimum [38]." In this research, genetic algorithms alone are used to optimize the problem. The following sections detail viable alternatives, and weigh their pros and cons against those of genetic algorithms, emphasizing the single-objective aspects of each method.

2.3.1 Parallel Systems. Liu et al. discusses parallel systems in their article on Earth observation satellite optimization [38]. The idea of parallel systems is to improve both a real system and a model of that real system through interactions between the two. The model can be improved by comparing results to the real system, and the real system is improved as weak areas are exposed through the model's evaluation. The models can be mathematical equations or virtual simulations. Experiments that use these models take into account all relevant system factors, both internal and external to the system, and its environment [38]. It is worth noting that these models may lose some of the unpredictability inherent in the real world, and can never be a perfect replica of the real system.

In the case of a space system, the "real" system is out of reach (in space) and so can be represented by ground station data for experimentation purposes [38]. This kind of data is what we will use to evaluate and improve our models.

2.3.2 Genetic Algorithms. An understanding of genetic algorithms is essential to this work, since this is the optimization algorithm used throughout. Genetic algorithms are meant to simulate biological evolution and Charles Darwin's theory of natural selection [41, 42]. Just like in nature, this process should theoretically yield the best results and shed the worst results in real-world problems that involve many variables and unpredictable situations. Some common terms are taken from Arora's textbook, *Introduction to Optimum Design* [41], and are summarized in Table 2.4:

Population	Current design vectors being tried
Population size	Number of designs per generation
Generation	Solutions for designs in one population
Iteration	One calculation within a generation
Tolerance	Sensitivity of the cost function
Chromosome	Design point
Gene	Value of a design variable

Table 2.4: GA Definitions [41]

Population. The set of design points at the current iteration, representing a group of designs as potential solution points. The *population size* is the number of designs in a population.

Generation. A calculation in the genetic algorithm, having a population of a given size (population size) that is manipulated to find the best function value. This may consist of multiple *iterations*, which are defined by specific values of design variables.

Tolerance. The smallest change in value that is considered significant. A function tolerance refers to the smallest change in the cost function between generations, and a constraint tolerance refers to the greatest constraint violation that is acceptable.

Chromosome. A synonym in genetic algorithms for a design point. This design can be feasible or infeasible, and contains values for all the design variables of the system.

Gene. A scalar, valued component of the design vector (the value of a particular design variable).

There are three parts to most function optimization problems, including genetic algorithms. These are the objective function, the variables, and the constraints. The objective function represents the value we want to minimize (or maximize). The variables represent resources or other factors affecting the objective function. The constraints restrict the solution vector to plausible values of the variables [42].

The first step in solving any problem is representing the problem. In genetic algorithms, this is done through encoding. Encoding techniques listed by Dr. Te-

jas Patalia include binary encoding, permutation encoding, value encoding, and tree encoding; the technique used will be problem-dependent. The process of encoding yields strings of values (the chromosomes) that represent some characteristic of the problem, and is the most difficult part of setting up the genetic algorithm [42].

The objective function in genetic algorithms is known as the fitness function, and it "defines the relative importance of a design [41]." The fitness function is used to select chromosomes, and is the judge of which values are passed along and which values are thrown out [42].

"The basic idea of a genetic algorithm is to generate a new set of designs (population) from the current set such that the average fitness of the population is improved [41]." Genetic algorithms have three processes that are iterated upon to achieve this. Reproduction, or selection, copies "good" chromosomes and uses them to replace "bad" chromosomes, keeping the population size constant. There are many methods of reproduction, which we will not detail in this paper [42]. Crossover is the next process, during which a pair of randomly selected "mating strings" exchange a few genes [41,42]. The third process is mutation, which can be thought of as a random bit flip that can prevent premature loss of genetic material [41]. A genetic algorithm flowchart is shown in Figure 2.6.

Genetic operators like crossover and mutation make GAs unique from other evolutionary computation techniques. Because of the chromosomes sharing information with each other, the entire population moves as a group toward one optimal area [43]. The genetic operators allows the GA to mutate (just like in natural selection).

Patalia concludes that genetic algorithms are more reliable, stronger, and more robust than any other traditional heuristic method for function optimization [42]. However, genetic algorithms have a couple disadvantages, such as requiring massive calculations and no guarantee of finding a global solution. These may be overcome by using parallel computers, multiple runs of the algorithm, and longer run times [41]. All in all, GA seem to be an effective optimization method for the problem at hand.

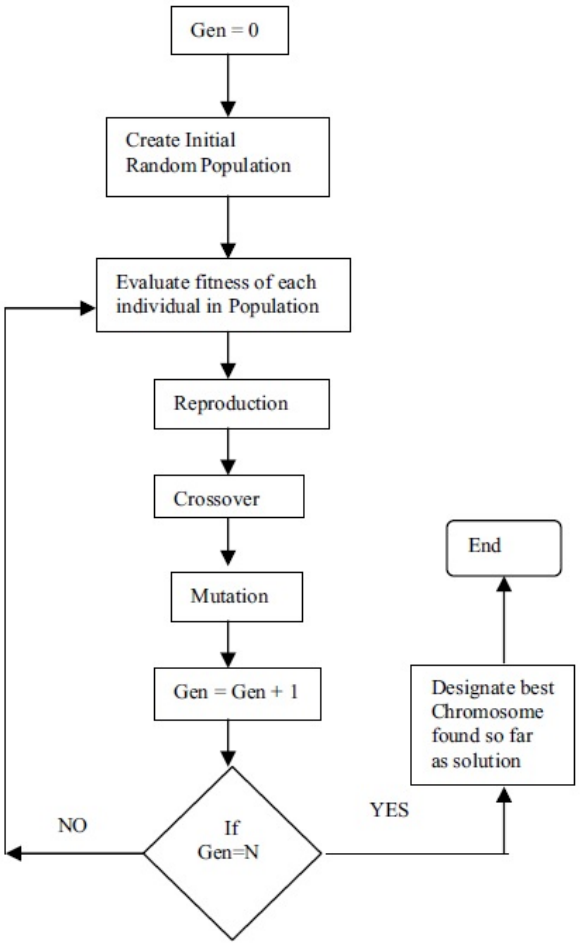


Figure 2.6: GA Flowchart [42]. The stopping condition for GAs can be a desired maximum number of generations as shown here, but could also be given a function tolerance value. If the cost does not change by more than the function tolerance value between generations, the GA will be considered finished. The GA used in this work utilizes a maximum of 50 generations, as well as a cost function tolerance of $10,000.

2.3.3 Particle Swarm Optimization. Particle swarm optimization (PSO) uses a population, referred to as a swarm, made up of individual members, or particles, to search for the optimal solution to a function. Leaders are used to guide the search, and each particle is assigned a velocity in a particular direction. It simulates a flock of birds looking for food [44]. Below are some common terms from Margarita Reyes-Sierra and Carlos Coello Coello [44], and are summarized in Table 2.5:

Swarm. Population of the algorithm.

Swarm	Algorithm population
Particle	Member of swarm; potential solution
Leader	Particle that guides others toward better solutions
Velocity vector	Direction of improvement
Inertia weight	Impact of previous velocities on current velocity
Learning factor	Attraction of a particle toward success

Table 2.5: PSO Definitions [44]

Particle. Individual member of swarm, representing a potential solution to the problem being solved. The position of a particle is determined by the solution it currently represents.

Leader. Particle that is used to guide another particle towards better regions of the search space.

Velocity (vector). This vector drives the optimization process by determining the direction in which a particle needs to "fly" to improve its current position.

Inertia weight. Controls the impact of the previous history of velocities on the current velocity of a given particle.

Learning factor. Represents the attraction that a particle has toward either its own success or that of its neighbors.

In PSO, "changes to the positions of the particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals [44]." Some merits of PSO over GA are found in Arora.

> [PSO] has fewer algorithmic parameters to specify compared to GAs. It does not use any of the GAs' evolutionary operators such as crossover and mutation. Also, unlike GAs, the algorithm does not require binary number encoding or decoding and thus is easier to implement on the computer. PSO has been successfully applied to many classes of problems... [41]

Reyes-Sierra et al. also highlights some of the differences between PSO and other evolutionary algorithms. Where GA uses three mechanisms (see section 2.3.2), PSO uses only two. There is no explicit selection function, and no offspring generation. PSO uses leaders to guide the search, and has an operator that gives particles a

velocity and direction. "Changes to the position of particles...are based on the social-psychological tendency of individuals to emulate the success of other individuals [44]." An algorithm for PSO is depicted in Figure 2.7.

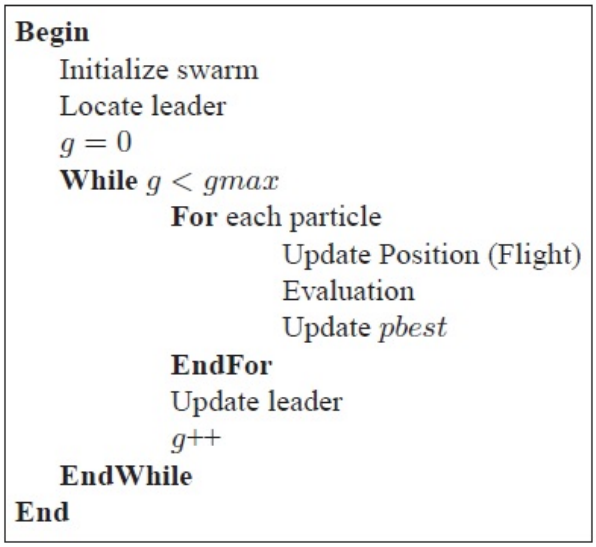


Figure 2.7: PSO Algorithm [44]

The downside to PSO is that it, like GA, may prematurely converge on a non-optimal solution. It is not guaranteed to find an optimum from an arbitrary initial state [44].

GAs and PSOs are the two most well-known heuristic optimization methods. Other methods include simulated annealing, tabu search, and threshold accepting. Because this is a nonlinear problem, there is no deterministic solution, and because the problem is discontinuous, there is no way to prove a solution is a global optimum. A direct search method is necessary, and no initial guess is provided. This thesis, and Maj. Thompson's work, make use of genetic algorithms, which suit all the needs of the problem at hand.

2.4 STK-Matlab Interface

By creating a Matlab script that generates a scenario and objects in STK automatically, some tedious tasks can be consolidated into loops, saving time for the user.

Anything that can be done in STK can be accomplished via a Matlab command. The hard part is figuring out how the commands work, since this is not well-documented. A Google search yields a few sparse examples of code, and the documentation in STK's programming help is written for languages such as C rather than Matlab. Navy LT James Sales created an STK library script in Matlab, which has been quite useful, and which Maj. Thompson and I have added to in our efforts to create a scenario for small and large satellite constellations. Our use of and amendment to the STK library script also aims at completing some of the future work mentioned in LT Sales' thesis [45].

2.5 Optics

A minimal understanding of optics is the last necessary piece of information for this problem. The algorithm developed during this research uses a constraint that the target must be visible according to the National Imagery Interpretability Rating Scale (NIIRS) level 3, which John Irvine's overview of NIIRS [46] qualifies as follows: Identify the wing configuration (e.g., straight, swept, delta) of all large aircraft Identify radar and guidance areas at a SAM site by the configuration, mounds, and presence of concrete aprons Detect a helipad by the configuration and markings Detect the presence/absence of support vehicles at a mobile missile base Detect trains or strings of standard rolling stock on railroad tracks (not individual cars)

The equation for determining the NIIRS level of a target, and other variables that need to be determined, are described in Chapter 3.

One more important value to optics is the f-number, which is simply the ratio of the focal length to the diameter. As a point of reference, the telescope on Landsat 7 has an f-number of 6.0 [47].

2.6 Applications

Disaggregation is applicable to many types of systems, provided the concept is applied in an appropriate manner. In particular, Earth observation payloads that require maximum Earth coverage could benefit from the effects of disaggregation. As reported in a recent article by Debra Werner in the summer of 2013, the Visible Infrared Imaging Radiometer Suite (VIIRS) on the Suomi National Polar-orbiting Partnership weather and climate satellite detected rising temperatures in Japan's Mount Sakurajima, warning operators of impending volcanic activity 14 hours before its eruption near Kagoshima City. Although it is unusual to catch such an event on a single instrument, this event highlighted the possibility of using such thermal imagery to predict volcanic activity and other environmental disasters. But William Straka cautioned, "If you rely on one source of information too much, you end up with a case of false detection [48]." It would be necessary to implement a whole system-of-systems dedicated to the volcano forecasting mission.

Similar to volcano forecasting is weather prediction. Weather satellites have many sensors, and global coverage is necessary to get the full picture on impending weather in even a small region. The Joint Polar Satellite System (JPSS) is to be the follow-on to the National Oceanic and Atmospheric Administration's (NOAA) Polar-orbiting Operational Environmental Satellites (POES), to provide global monitoring capability for the United States. Its concept of operations can be seen in Figure 2.8. JPSS will utilize the Common Ground System (CGS) that is already in use by many international weather and environmental sensing missions [49]. This system is under threat of being delayed because of withheld funding. If the delay happens, there will be an observational data gap in late 2016 to early 2017 as POES expires and JPSS awaits operations [50].

The US Army has aspirations of a responsive imaging system to get imagery data to warfighters on the ground, and has been exploring some options resembling disaggregated constellations to provide the kind of coverage necessary for these op-

Figure 2.8: The JPSS Concept of Operations [49]

erations. This kind of reconnaissance is an example of many military interests in disaggregation.

2.7 Summary

This chapter has outlined the existing work regarding disaggregation and optimization. The next chapter will detail the specific models used in this thesis, and describe the setup of the algorithm.

III. Methodology

The Army has exhibited a need for responsive imaging systems with versatile coverage, by way of the Kestrel Eye and NanoEye programs that are currently under development. This research uses a genetic algorithm to determine the number of and parameters for imaging satellites in a mixed constellation, thereby simulating a potential imaging system and allowing its results to be analyzed. An illustration of the scenario studied in this research can be seen in Figure 3.1.

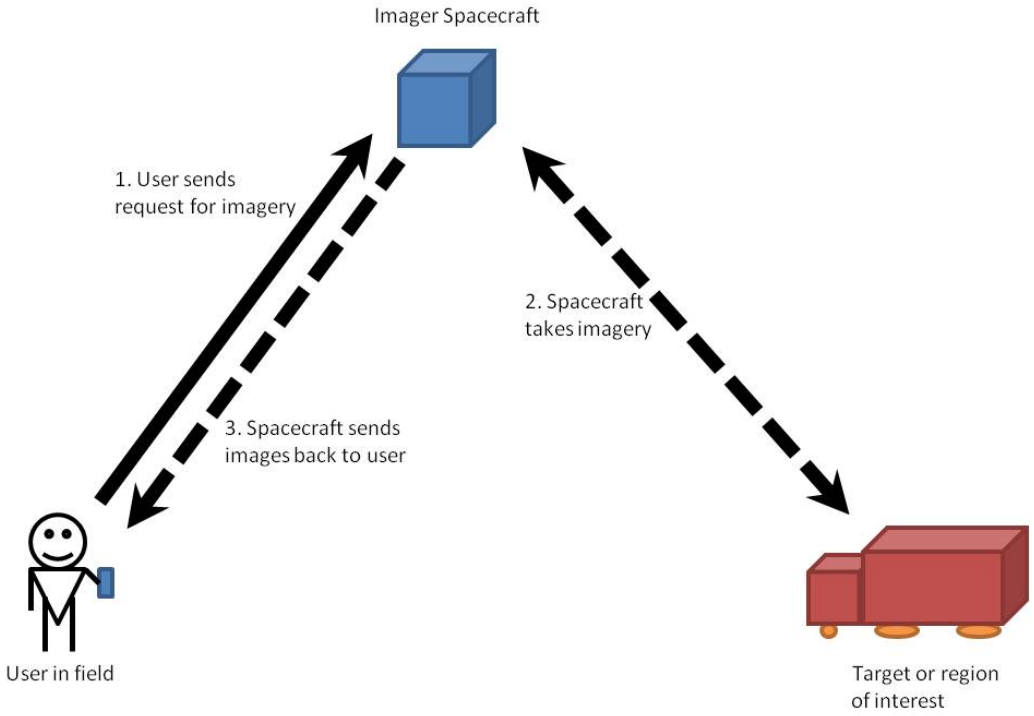


Figure 3.1: Army Responsive Space Concept. The user sends a request from a handheld device for an image from a particular region. The spacecraft takes the desired images as it flies over the region or target, and sends them back to the users device. The dashed line represents the spacecrafts actions, and the solid line represents the users actions.

3.1 Problem Statement

The scenario is based on recent Army programs such as Kestrel Eye, NanoEye, and SATS, which have similar missions. According to a small business innovation research (SBIR) report for Kestrel Eye, the Army would like to have access to persistent imagery coverage tasked by "lower echelons," and have the imagery directly

downlinked from the satellite to the tasking forces, for "battlespace awareness" purposes. The Army is willing to compromise imaging resolution to get delivery within a few minutes of tasking [31]. This is quantified in the model as a minimum NIIRS level of 3, and a desired response time of 5 minutes. The region of conflict is assumed to be the Middle East. With the problem thus stated, the model is implemented.

This model assumes two types of spacecraft in different orbital planes. The types of spacecraft are large and small, which implies different cost and sensor performance considerations based on weight and sensor aperture diameter. The number of orbital planes containing each type of satellite and the number of satellites in each plane are variable. This arrangement simulates a mixed constellation and attempts to determine the most cost-effective combination of assets in order to get persistent target coverage. Mixed constellations combine exquisite systems currently in use with less capable, shorter-life, targeted capability payloads that still accomplish requirements [1]; the "large" satellites might be thought of as being augmented by the "small" satellites. Large here refers to spacecraft between 300 and 5200 kilograms, while small refers to spacecraft under 500 kilograms. The size of these spacecraft is based on optic sensor aperture diameter, which is derived from the mass using historical values as scaling factors.

Twenty-five targets are placed in five locations, in a pattern around a Middle East location using a Matlab script called targets.m (see Appendix D), and represent user requests from a common area of operation. User requests will come in randomly and will likely reside in a similarly sized region of interest. The pattern of targets used can be seen in Figures 3.2 and 3.3. A similar pattern of targets around Ohio is used to investigate the effects of higher latitude targets on the optimal constellation, and the code also includes a target deck in North Korea for future study of various target decks.

The scenario time is 48 hours, long enough to ensure against a surge of passes with a long gap between them, but short enough to keep the calculation speed down.

The mission objective is to image all of the requested targets with an effective response time, which is entered as a constraint and can be varied on different runs of the program. This research optimizes a constellation for two specific target decks located in the Middle East and in Ohio; this algorithm would need to be run for current existing targets in a real-world situation. The time the GA takes to optimize the constellation is very long, but can be shortened by using parallel processing and faster computers.

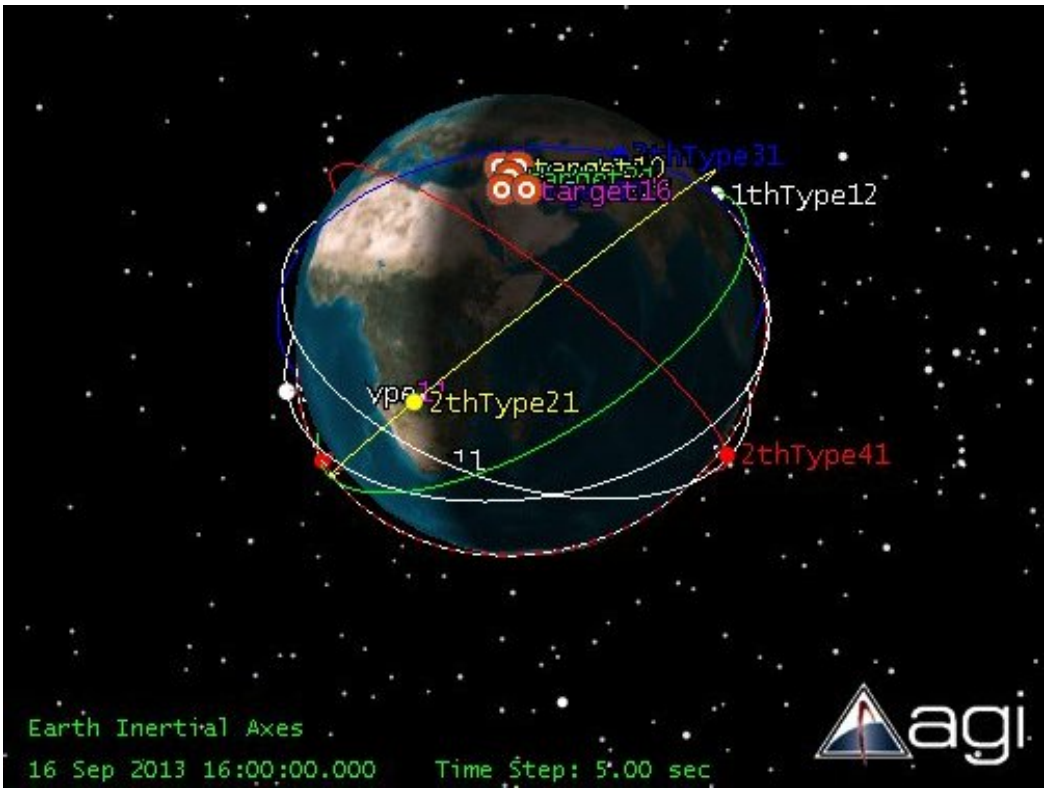


Figure 3.2: Assumed targets. 25 targets are placed at edges of an assumed region of interest in the Middle East. Each target can be assigned a different NIIRS value within the Matlab program, constraint.m. Only five targets are visible since the pattern places targets on top of each other to represent, for example, five aircraft at an airfield (multiple targets in a small area).

3.1.1 Assumptions. The scenario results depend on the following assumptions: responsive launch vehicle available, capable ground infrastructure, ideal onboard subsystems (propulsion and attitude determination and control subsystem) capable of station keeping for the duration of the mission lifetime, clear visibility of

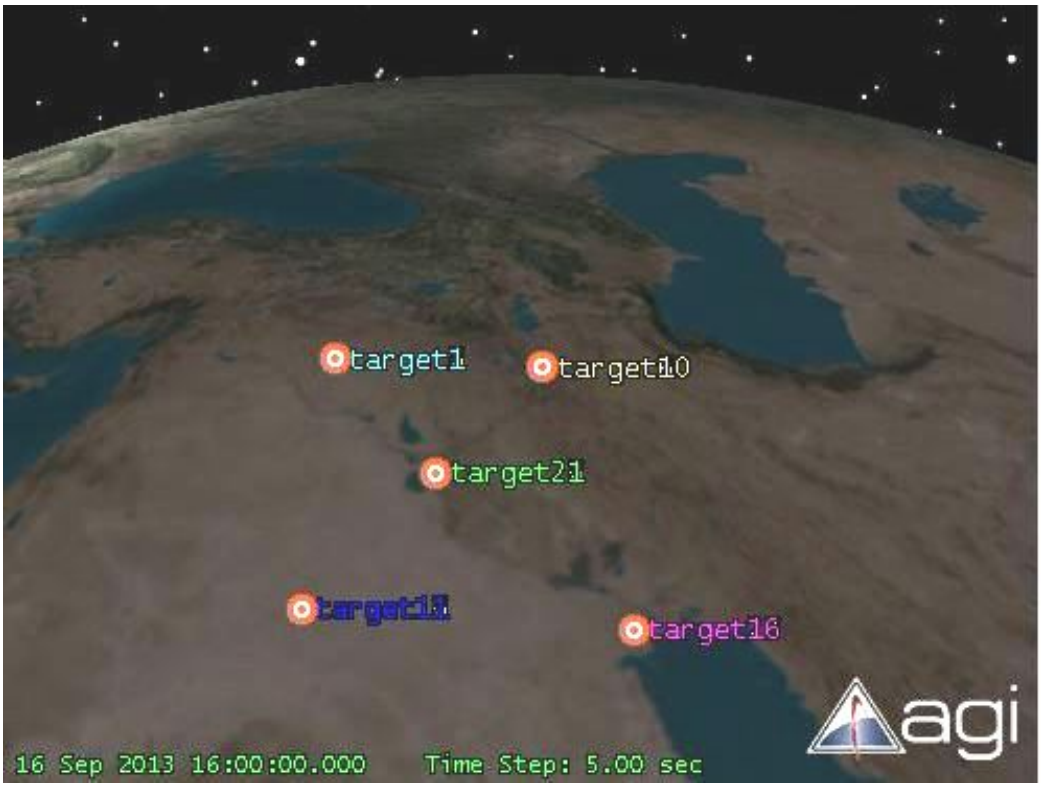


Figure 3.3: Assumed targets. 25 targets are placed at edges of an assumed region of interest in the Middle East. Each target can be assigned a different NIIRS value within the Matlab program, constraint.m.

targets (no weather obstructions), and costs in fiscal year 2010 dollars (FY2010$). The response time in this thesis refers to the maximum amount of time any target is out of coverage. This thesis does not account for downlinking data to a groundstation. The orbits of spacecraft are assumed to be circular. The cost models used also carry assumptions that are enumerated in Wertz [4], and are summarized here.

The Unmanned Space Vehicle Cost Model, version 8 (USCM8), derives cost estimating relationships (CER) from a total of 44 satellites using statistical regression techniques to support parametric cost estimates of unmanned, earth-orbiting space vehicles with a communications payload. This research uses the National Aeronautics and Space Administration (NASA) Instrument Cost Model (NICM) for optical Earth-orbiting payloads in conjuction with USCM8, and it should be noted that combining models in this way incurs further risk in accuracy of results. For smaller spacecraft,

Parameter	Units	Bounds (Large)	Bounds (Small)
# Planes	-	0-10	0-20
Sats/Pl	-	0-10	0-20
Truan	deg	0-360	0-360
RAAN Inc	deg	0-360	0-360
Alt	km	200-5000	200-5000
Incl	deg	30-90	30-90
Ecc	-	0	0
RAAN	deg	0-360	0-360
ArgPer	deg	0-360	0-360
M	deg	20	20
Diam	m	0.1445-0.3774	0.0471-0.1612

Table 3.1: Design vector parameters. The GA varies the design vector within the bounds shown. Most parameters are ranges but some are simply values; the user may desire to vary these parameters in subsequent experiments but they are held constant for this research.

the Small Spacecraft Cost Model (SSCM) is used, which is meant for spacecraft under 500 kilograms and does not account for a specific payload type. This model does not distinguish between non-recurring and recurring costs, but historical values were used to determine the breakout of each piece within the code (although that is inconsequential since they are simply combined immediately) [4]. Four cost models can be found in Table 3.2 on page 51.

3.1.2 Design Variables. The genetic algorithm will determine a design vector that resides within specified bounds. The design variables chosen for this experiment can be divided into three subcategories: Walker parameters, orbital parameters, and sensor parameters. The Walker parameters are number of planes (# planes), number of satellites per plane (sats/pl), true anomaly spread (TruAn), and RAAN increment (RAAN inc). The orbital parameters are altitude (alt), inclination (incl), eccentricity (ecc), right ascension of the ascending node (RAAN), argument of perigee (ArgPer), and mean anomaly (M). The sensor parameter is aperture diameter (diam). These values are summarized in Table 3.1.

The bounds were determined based on a LEO mission for sensors of the sizes specified in the USCM8 and SSCM cost models, discussed in the previous section. The Walker parameter bounds are representative of the maximum number of satellites,

$(x_1 * x_2)$, to be put in orbit. Number of planes and number of satellites per plane were used as design variables instead of simply the number of satellites because it is easiest to calculate the number of satellites as a product, to avoid a potential divide by zero error that would result from dividing the number of satellites by the number of planes, should the number of planes be zero. This research varied the number of planes and sats/plane over different runs, to try to determine what bounds are best to use so as not to restrict the GA too much.

LEO works well for localized coverage rather than global coverage [4], and of course provides better signal-to-noise ratio (SNR) and resolution. Therefore the program is bounded to LEO altitudes. Note should be taken that, although the model allows altitudes up to 5000 km, Van Allen radiation makes orbits above about 1000 km less feasible as there is a need for radiation hardening. Inclination is bounded between 30 and 90 degrees based upon our Middle Eastern target region location, since we know that near-equatorial orbits will not be useful, and all other angles are bounded from 0 to 360 degrees. Often in the code, machine zero $(1 * 10^{-6})$ is used in place of zero.

The USCM8 Theoretical First Unit (TFU) model is for spacecraft with a bus mass between 288 kg and 7398 kg, while the Non-Recurring Engineering (NRE) model is for spacecraft between 114 kg and 5127 kg. To use both these models, the bounds are from 288 kg to 5127 kg, and from there the corresponding sensor diameter is computed using Equations 3.1 and 3.2 [4].

$$\begin{aligned} m_b &= m - W_i \\ &= W_i/0.31 - W_i \\ W_i &= m_b/2.2258 \end{aligned} \tag{3.1}$$

Use this weight in the equation for aperture ratio (R); calculate R first with constant K=1, recalculate with K=2 if $R < 0.5$

$$R = \sqrt[3]{W_i/(KW_o)}$$
$$A_i = R * A_o \tag{3.2}$$

where m_b = bus mass, m = dry mass, W_i = payload mass, R is aperture ratio, A_i is the required sensor aperture diameter, and A_o and W_o are given by the reference payload aperture and mass as 0.18 m and 250 kg, respectively, for a reference imaging spectoral instrument found in Table 17-7 of Wertz [4].

There are many optical parameters that are assumed to be constant. These include effective focal length, assumed to be 10 meters (Kestrel Eye is also 10 meters [31], while Ikonos-2 is 8 meters [51]), FOV along-track (assumed to be same as cross-track), pixel size (assumed to be 30 microns), line rate (unused), time delay integration (unused), data quantization (unused), and data compression technique (unused) [51].

3.2 The Model

The main wrapper, or program that runs all aspects of the algorithm, for this simulation was created by Major Robert Thompson. Coupled code includes the constraint and cost functions for the genetic algorithm. Within the constraint function is the script to calculate coverage using the COM interface to STK. For this scenario, coverage is desired to be greater than 95 percent. The NIIRS constraint is level three, to reflect the intention the Kestrel Eye SBIR [31]. If higher resolution is desired (potentially at the cost of a longer time), one needs only to change the parameters in the model. The script that interacts with STK was created using functions from LT James Sales' STK library [52], to which we added a few functions more specific to our needs. The full code can be found in Appendix D.

The wrapper program, image_ga.m, contains the bounds and the call to the genetic algorithm function in Matlab. Options used include a population size of 20,

a maximum of 50 generations, a stall generation limit of 10, and a function tolerance of ten, which equates to ten thousand dollars. Most of these terms were defined in Chapter 2; the stall generation limit refers to the number of generations over which cumulative change in fitness function value is less than the function tolerance [53]. The output function of the GA was also modified to output the state at each iteration, so that an early termination due to a runtime error would not result in loss of data. The constraints are specified in constraint.m and restrict the solver to a National Image Interpretability Rating Scale (NIIRS) for all targets of three, a coverage of at least 95% of targets over the scenario period, and a maximum revisit time of a given number of minutes. The NIIRS constraint can be input as a vector with length equal to the number of targets, giving each target a different NIIRS constraint.

The NIIRS value is calculated in Sensor_performance.m, an amended version of Maj. Thompson's original work. It uses slant range to targets (taken from the STK scenario) in addition to altitude, since the signal to noise ratio is defined by the linear distance between the target and the sensor. The equations are all taken from [54]. The NIIRS value is calculated by the General Image Quality Equation (GIQE):

$$NIIRS = c_0 + c_1 * \log GSD + c_2 * \log RER + c_3 * H + c_4 * \frac{G}{SNR} \tag{3.3}$$

Each parameter is defined and determined as outlined in Appendix B.

The field of view (FOV) is not used in calculating the NIIRS value, but it can be used in calculating the necessary conic half angle for the sensors that are created in the scenario as $FOV/2$.

$$\begin{aligned} IFOV &= \frac{d}{f} \\ FOV &= IFOV * N_m \end{aligned} \tag{3.4}$$

where IFOV is the instantaneous field of view (radians), and N_m is the number of focal plane array pixels, assumed to be 256. The sensor conic half-angle is then simply

half the FOV, converted to degrees. This calculation is included for reference, but for simplicity the program assumes a conic half-angle of 10 degrees for the staring sensor and 0.28 degrees for the targeted sensor as standard values [4]. For reference, GeoEye's Ikonos-2 is a large spacecraft with a field of view of 1.19 degrees, cross-track [51], so the 10 degree assumption is an over-estimation. Slewing is accounted for in the STK simulation by using targeted sensors with a max slew rate of 1 degree per second. The assumed half-angle for the targeted sensor comes from:

$$FOV = \frac{1.22 * \lambda}{D} \tag{3.5}$$

for a single pixel, so for a state-of-the-art system with 8000 pixels,

$$\begin{aligned} &= \frac{1.22 * 0.5 * 10^{-6} m}{1m} * 8000 pixels \\ &= 0.0049 rad = 0.2807 deg \end{aligned} \tag{3.6}$$

The sensors are all assumed to have an elevation angle of 90°, or nadir-pointing. The call to Sensor_performance.m is within ga_scenario.m.

The coverage constraint is calculated with gascenario.m, and takes the bulk of the computing time. This function utilizes the STK library and COM interface. Within this program is a command that turns off the STK visualization. This feature was useful in speeding up computation time and reducing run-time errors.

The cost function, cost.m, calculates the cost in 2010 thousands of dollars for the entire system. This is the value the genetic algorithm attempts to minimize. The cost is based off the USCM (large spacecraft) and SSCM (small spacecraft) cost models. Four cost models from Wertz [4] can be seen in Table 3.2; USCM and SSCM are used for their combination of simplicity and thoroughness as a standard approximation. All models are imperfect and introduce errors, and the cost models are approximations that carry many assumptions and should be compared to solutions

using the same cost models and compounding the same errors. In addition, launch and ground operations costs are not included. Solutions found by this model will be relative and should not be compared to real-world system costs.

The USCM8, SSCM, and NICM equations from Table 3.2 are implemented in the cost function. The QuickCost would be an easy calculation, but would lump the small and large satellite types together, whereas it is important to this research to keep those values separate. It is worth noting that these models account for launch integration costs but not launch vehicle costs. In the real world, one might use, for example, one rocket per plane for up to eight satellites. Maj. Thompson is working on models for also estimating launch vehicle costs, and further work should be done to include launch estimates in this optimization algorithm.

Major Thompson's work so far has been based on calculating global coverage with a Walker constellation, whereas this work focuses on a given target deck. The goal is no longer global coverage, but coverage of important targets with a given response time requirement. By investigating the effects on specific target decks, one must remember that each solution is unique to the assumed target deck. Another difference between this work and Maj. Thompson's is the use of targeting reconnaissance sensors, rather than push broom or whisk broom surveillance sensors, adding an element of flexibility to the constellation's sensing abilities.

3.3 Validation

The cost model was compared with results from the New SMAD [4].

The sensor performance model was validated by comparing results with the Kestrel Eye specs in the Army SBIR [31] for a satellite with similar size and range. After generating results, the ratio of the assumed focal length, 10 meters, and the diameter of most solutions, 0.1 meters, were found to give an f-number of about one hundred, yielding unrealistically high NIIRS values. To remedy this, the assumed

SME-SMAD Cost Models, FY2010$			
USCM8 Non-recurring Subsystem CERs in FY2010 Thousands of Dollars			
Element	*Equation*	*Variable*	*Ref*
1. Spacecraft bus	110.2*X	X=Spacecraft weight (kg)	Table 11-8
2. Payload	1163*M^.426*P^.414*DL^.375	M=instrument total mass (kg), P=Max instrument Power (W), DR=total data rate (kbps)	Table 11-14, Optical Earth-Orbiting Payload NASA Instrument Cost Model (NICM)
3. Integration, Assembly, and Test	0.195*X	X=Spacecraft bus + Payload Non-recurring Cost ($k)	Table 11-8
4. Program Level	0.414*X	X=Space vehicle and IAT Non-recurring cost	Table 11-8, non-comm satellite
5. Aerospace Ground Equipment (AGE)	0.421*X^.907*2.244	X=Spacecraft bus Non-recurring cost ($k)	Table 11-8, non-comm satellite
USCM8 Spacecraft Bus Recurring T1 CERs in FY2010 Thousands of Dollars			
Element	*Equation*	*Variable*	*Ref*
1. Spacecraft bus	289.5*X^.716	X=Spacecraft weight (kg)	Table 11-9
~~2. Payload~~	~~1163*M^.426*P^.414*DL^.375~~	~~M=instrument total mass (kg), P=Max instrument Power (W), DR=total data rate (kbps)~~	~~Table 11-14, Optical Earth-Orbiting Payload NASA Instrument Cost Model (NICM)~~
3. Integration, Assembly, and Test	0.195*X	X=Spacecraft bus + Payload Non-recurring Cost ($k)	Table 11-9
4. Program Level	0.414*X	X=Space vehicle and IAT Non-recurring cost	Table 11-9, non-comm satellite
5. Aerospace Ground Equipment (AGE)	0.421*X^.907*2.244	X=Spacecraft bus Non-recurring cost ($k)	Table 11-9, non-comm satellite
SSCM Total Non-recurring Cost (development plus one protoflight unit)			
Element	*Equation*	*Variable*	*Ref*
1. Spacecraft bus	1064+35.5*X^1.261	X=Spacecraft dry weight (kg)	Table 11-11
2. Payload	.4*X	X=Spacecraft Bus Total Cost ($k)	Table 11-11; could also use NICM?
3. Integration, Assembly, and Test	0.139*X	X=Spacecraft Bus Total Cost ($k)	Table 11-11
4. Program Level	0.229*X	X=Spacecraft Bus Total Cost ($k)	Table 11-11
5. Launch & Orbital Operations Support (LOOS)	.061*X	X=Spacecraft Bus Total Cost ($k)	Table 11-11
6. Ground Support Equipment (GSE)	0.066*X	X=Spacecraft Bus Total Cost ($k)	Table 11-11
QuickCost Non-recurring plus Recurring (T1) CER in FY2010 Dollars			
1. Space Vehicle for Unmanned Robotic Mission	2.829*M^0.457*P^0.157* 2.718^(0.171*%D)* 2.718^(0.00209*L)* 2.718^(1.52*%N)* 2.718^(.258*Planetary)* 1/(2.718^(.0145*(Y-1960)))* 2.718^(.467*C)*1/(2.718^(.237*T))	M=dry mass instruments + s/c (kg); P=Power (W); %D=Percentile relative to state-of-the-art (0-1); L=design life (months); %N=percent new tech (0-1); Planetary=0 for Earth-orbital, 1 for planetary; Y=Authority to proceed 4 digit calendar year; C=Percentile relative to "average" instrument complexity (0-1); T=team experience (1-4)	Table 11-12

Table 3.2: Cost Models [4]. In USCM8 TFU, the payload value is crossed out since it is already accounted for in the NRE value.

focal length is changed to six times the design diameter, or about 0.6 meters. One run is completed with this parameter to give an example of more realistic results.

The overall results were checked for common sense; for example a shorter revisit time should require more satellites, and a higher cost should be due to a greater number of satellites or larger satellites (greater diameter).

An example calculation using a reference solution is completed in Appendix C.

3.4 Implementation

The solutions were generated using the following steps:

1. Input desired target locations in targets.m and run the program to generate STK target deck file
2. Check filepaths in gascenario.m and image_ga.m
3. Input desired constraint values for percent coverage, NIIRS value, and maximum revisit time in constraint.m
4. Run image_ga.m
5. Solution is saved in specified file; check exitflag and run x through constraint.m to see actual revisit time

The first step was to generate requested targets. This task was accomplished using targets.m, which generates 25 evenly-spaced targets in the region of interest (see Figure 3.3). Each target may represent a group of targets, such as a fleet of aircraft or tanks, and may be assigned a unique required NIIRS value within constraint.m. The next step is to run the genetic algorithm, which minimizes system cost while meeting constraints: at least 95% target coverage, a NIIRS of at least the required value for all targets, and a certain maximum response time. The genetic algorithm must be run many times in order to yield significant results, since genetic algorithms are not guaranteed to produce the global minimum every time. Due to constraints on research time available, the number of runs completed for this thesis still does not

provide confidence that the algorithm ever reached a global minimum. This entire process is repeated for every case studied. Each case varies the revisit requirements. Varying the target deck is outside the scope of this thesis, however a few solutions were calculated with targets in Ohio to get an idea of the effects of different latitude targets.

To expedite the process, multiple computers were used to run trials of different response times simultaneously. Computers used included an AFIT HP laptop with an Intel Core Duo CPU and 4 GB of RAM with 64-bit Windows 7 and AFIT Dell desktops with AMD Athlon Dual-Core processors, 4 GB of RAM, and 64-bit Windows 7. The unique lines of code that must be checked before beginning each run are the response time and desired NIIRS values in constraint.m, the STK path in gascenario.m, and the filename used for saving data found in the main wrapper, image_ga.m (see Appendix D). The number of trials in the wrapper program can also be customized. A flowchart showing the steps of the algorithm is shown in Figure 3.4.

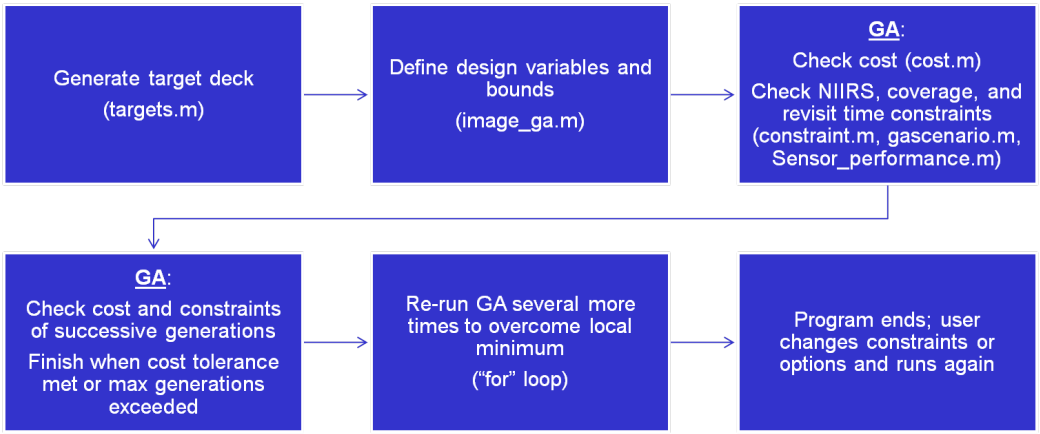


Figure 3.4: Algorithm flowchart. A flowchart showing the steps the algorithm takes to determine the optimal constellation parameters.

The runtime for the program is about 12 hours, and so a "for" loop was used to allow the program to run multiple experiments in a row. This way, the simulation could be left to run by itself and useful results could be gathered over several days.

After each run of the genetic algorithm, the results were saved along with a modifier of either "success" or "failure" to denote whether the GA ran to completion or was terminated early due to a runtime error in STK. The long runtime is due to successive access calculations in STK for each target. The STK portion of the algorithm takes up to 5 minutes per iteration. This may be improved upon in the future by using parallel access computations.

3.5 GA Outputs

After the program finishes, the results can be retrieved by opening the generated ".mat" and ".fig" files, which store the entire workspace and GA graphic for the associated run. The output of the genetic algorithm includes the cost in thousands of dollars (fval), the design vector (x), and an output structure that contains the status at finishing (output.message), the number of generations the algorithm took (output.generations), and the maximum constraint violation (output.maxconstraint). Common messages convey that the GA finished with all values within tolerances, and that the GA finished but was not able to find a point meeting all constraints. Only solutions of the first type contain any meaningful information.More detail about the constraint violation can be seen by running constraint.m with the final design vector. The exitflag is another important piece of information; 1 means there was no issue, -2 means constraints were not met, and 0 means that the maximum number of generations was exceeded. Results that violated constraints were not considered valid. If the user desires to see the entire population, 20 rows (for population size 20) of design vectors are available in the variable called population, and the corresponding costs can be found in the variable called scores.

Another output, selected as an option within the genetic algorithm, is the genetic algorithm graphic showing the best function value by generation, and the overall best solution. The upper plot shows how the GA starts with higher cost solutions and decreases the cost after each generation. An example of this is shown in Figure 3.5.

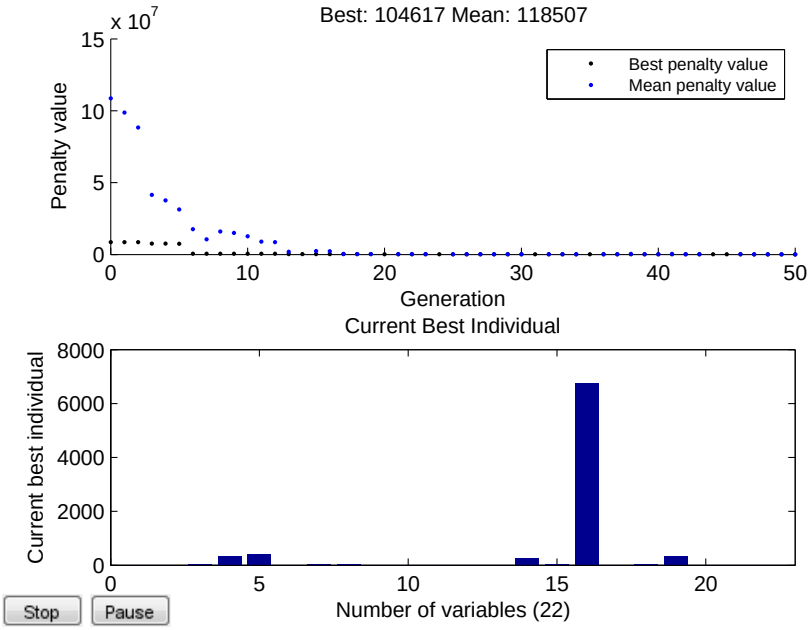


Figure 3.5: Example of GA graphic. This is the output plot for a 30 minute revisit time case using the Middle Eastern target deck. One can see that the GA finds high values at first and then zeros in on the best function value. The complete data can be found in Appendix E.

3.6 Summary

This chapter has established the design vector and equations used within the algorithm, and the flow of the algorithm for determining the optimal disaggregated imaging constellation for coverage of the Middle Eastern theater. The results will be discussed in Chapter 4.

IV. Results

The algorithm was run a total of approximately 40 times with a Middle Eastern target deck or with a higher latitude Ohio target deck. The solutions were screened for validity based on whether the algorithm successfully finished, and whether the solution was useful in comparison to prior solutions. The file targets.m (see Appendix D) also includes the code to generate a target deck in North Korea that was not run because it is similar in latitude to the Middle Eastern location. Many preliminary runs led to changes to make the model better and produce more accurate results. A maximum revisit time of two hours allowed familiarity with the process, and as the model improved the revisit time could be comfortably dialed down. Ideally, for the Army's mission, the time constraint should be decreased to less than ten minutes as was stated in the Kestrel Eye fact sheet [30]. This chapter outlines the best results from the simulation; the complete data for the cheapest and fastest solutions can be found in Appendix E, including GA plots and population tables.

4.1 Results of the Simulation

The results of the simulation are summarized in Table 4.1. As shown in the table, the number of satellites and cost of constellation are generally inversely proportional to the revisit time, as one might expect. A faster revisit time will require more satellites, and each satellite incurs a cost. Recall that the cost depends on size, which is a function of diameter. Also keep in mind the caveat that these solutions are relatively comparable, but do not represent actual costs. In many of the figures shown, a normalized cost is used to emphasize the relative nature of the solutions.

Although the trend between response time and cost or number of satellites seems intuitive, the precise relationship is more visible when plotted on a scatter chart (Figure 4.1, Figure 4.2).

Middle East				Population							
constraint	Revisit	NIIRS range	Cost ($B)	nplanes	sats/ plane	RAAN inc (deg)	true an spread (deg)	alt (km)	incl (deg)	RAAN (deg)	diam (m)
NIIRS=3				1	1	188	324.5	1462.7	31.4	214.5	0.1475
120min	103.955	4.85-7.23	0.43346	42	1	318.1	196.3	5898.1	88.5	92	0.0475
NIIRS=3				2	2	177.6	299.8	1326.9	33	76.3	0.1635
120min	87.93	6.47-7.5	1.1793	30	2	287.4	296.6	4259.2	77.3	140.5	0.0578
NIIRS=3				2	1	165.4	208.6	1353.8	54	219.8	0.1925
90min	65.9823	6.67-7.55	1.695	8	6	129.9	326	5500.7	58.9	164.6	0.1122
NIIRS=3				2	1	167.6	29	1085.9	78	335.9	0.1985
60min	28.6469	4.6-8.3	2.8391	28	5	70	334.8	5490.8	52	300.8	0.0843
NIIRS=3				6	1	101.8	107.1	892.7	84.5	187.3	0.1475
15min	11.5264	7.76-7.81	3.0619	38	6	62.3	34.3	5330.9	38.9	83.7	0.0951
NIIRS=3				1	8	168.9	269.9	295.2	41.9	138.8	0.1824
15min	1.8936	13.1-14.29	5.3905	40	7	63.6	131	628.5	50.5	164	0.1026
NIIRS=8				2	15	335.4	75.3	366.2	87.8	129.5	0.1629
30min	6.9735	10.4-11.3	7.9469	27	14	341.3	62.9	6805.9	34.1	339.9	0.0699
Ohio				Population							
constraint	Revisit	NIIRS	Cost ($B)	nplanes	sats/ plane	RAAN inc (deg)	true an spread (deg)	alt (km)	incl (deg)	RAAN (deg)	diam (m)
NIIRS=3				1	1	46.3	91	576.7	49.4	115	0.3319
90min	88.5	7.98-8.45	4.5753	9	9	139.3	36.5	4741.3	72.7	84.4	0.1237
NIIRS=3				1	1	112.1	265.2	759.3	46.8	52.7	0.1979
120min	74.7683	8.8-10	1.2029	4	5	220	286.2	6324.4	48.8	107.6	0.1248
NIIRS=3				1	1	212.1	217.1	1384.3	41.2	175.5	0.1456
60min	48.8	3.9-6.8	0.42348	8	4	115.8	76.4	5435.3	52.9	212.1	0.053
NIIRS=3				1	1	327.1	247.8	854.8	65.2	213.5	0.1644
60min	43.7	6.5-8.3	0.84296	7	8	10.2	322.5	5512.1	54.4	179.7	0.0702
NIIRS=3				1	2	332.9	121.9	1920.1	48.7	277.3	0.1981
30min	27.0975	4.09-4.58	1.1509	7	7	255.9	286.3	6725.6	38.4	66.8	0.0556
NIIRS=3				1	1	260.2	114.9	326.5	49.6	202.3	0.177
20min	12.888	8.8-13.07	1.5249	9	9	46.3	202.9	4777.2	44	174.5	0.0823
NIIRS=3				1	1	308.4	199.3	1792	53.5	102.3	0.2071
15min	13.6661	3.5-5.2	2.2471	8	7	246.7	59.6	6448.7	47.9	55.7	0.1276

Table 4.1: Summarized results of simulation. The design vector is split into two rows with the parameters for large satellites on top and for small satellites on the bottom. The number of satellites and cost of constellation are inversely proportional to the required revisit time. The seemingly abnormal altitudes compared to the diameters are attributed to the f-number of the optic, which will be further explained in a later section.

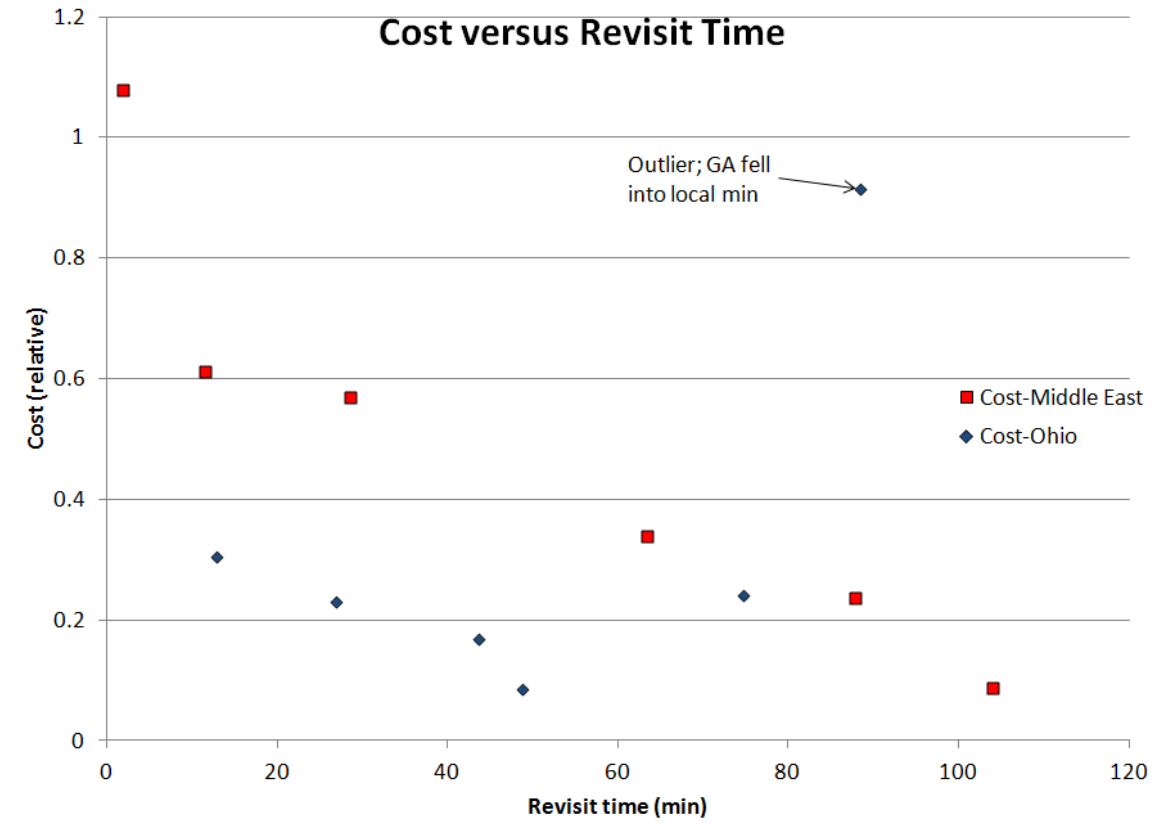


Figure 4.1: Cost versus Revisit Time. The cost has been normalized to $5 billion referenced to the cost model used to emphasize the purely relative nature of solution costs.

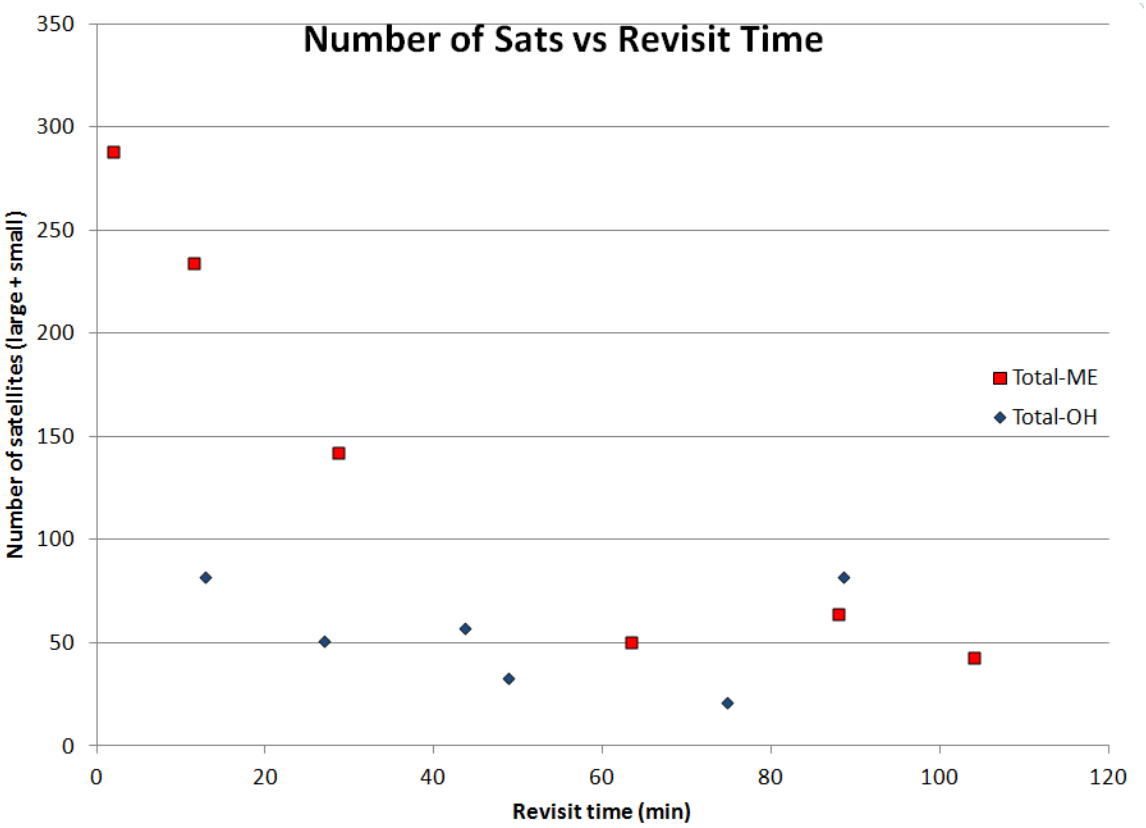


Figure 4.2: Number of Satellites versus Revisit Time

4.2 Refining Results

The raw data shows some trends forming, but we must also consider that a lower cost solution for a faster response time is also a solution for all slower response times (see Table 4.2). Taking this into account, the charts may be refined using the cheaper but faster solutions to make the relationships more accurate. The expensive constellation with a 90 minute revisit time (see Table 4.1 and Figure 4.1) is an instance where the GA fell into a local minimum and so was unable to find the global minimum. This effect was mitigated by refining the results (Table 4.2). We still cannot be sure that these solutions are global minimums, but they are the best results found so far.

Middle East					
Revisit-ME (min)	Cost-Middle East ($B)	Nsmall-ME	Nlarge-ME	Total-ME	$M/sat
104	$ 0.43	42	1	43	$ 10.08
87.93	$ 1.18	60	4	64	$ 18.43
63.4215	$ 1.70	48	2	50	$ 33.90
28.65	$ 2.84	140	2	142	$ 20.00
11.5264	$ 3.06	228	6	234	$ 13.09
1.8936	$ 5.39	280	8	288	$ 18.72
Ohio					
Revisit-OH (min)	Cost-Ohio ($B)	Nsmall-OH	Nlarge-OH	Total-OH	$M/sat
48.8	$ 0.42	32	1	33	$ 12.83
43.7	$ 0.84	56	1	57	$ 14.79
27	$ 1.15	49	2	51	$ 22.57
12.8888	$ 1.52	81	1	82	$ 18.60

Table 4.2: Refined results of simulation. Same as Table 4.1, but more expensive, slow revisit times are removed. Again, costs are shown for comparison but do not reflect actual prices.

The trendlines seen in Figures 4.3 and 4.4 were generated by displaying the coefficient of determination, R^2, and choosing the type of trendline (exponential, polynomial, power, etc.) that produced the R^2 value closest to one. The relationship is quantified by the equation displayed on the trendline. Confidence in these trends can be further improved as more data points are collected, checked against prior solutions to eliminate obvious local minimums, and added to the chart.

Several NIIRS values are higher than 9, which is infeasible for any orbital imager, but is attributable to the f-number of the notional system, described later. That issue

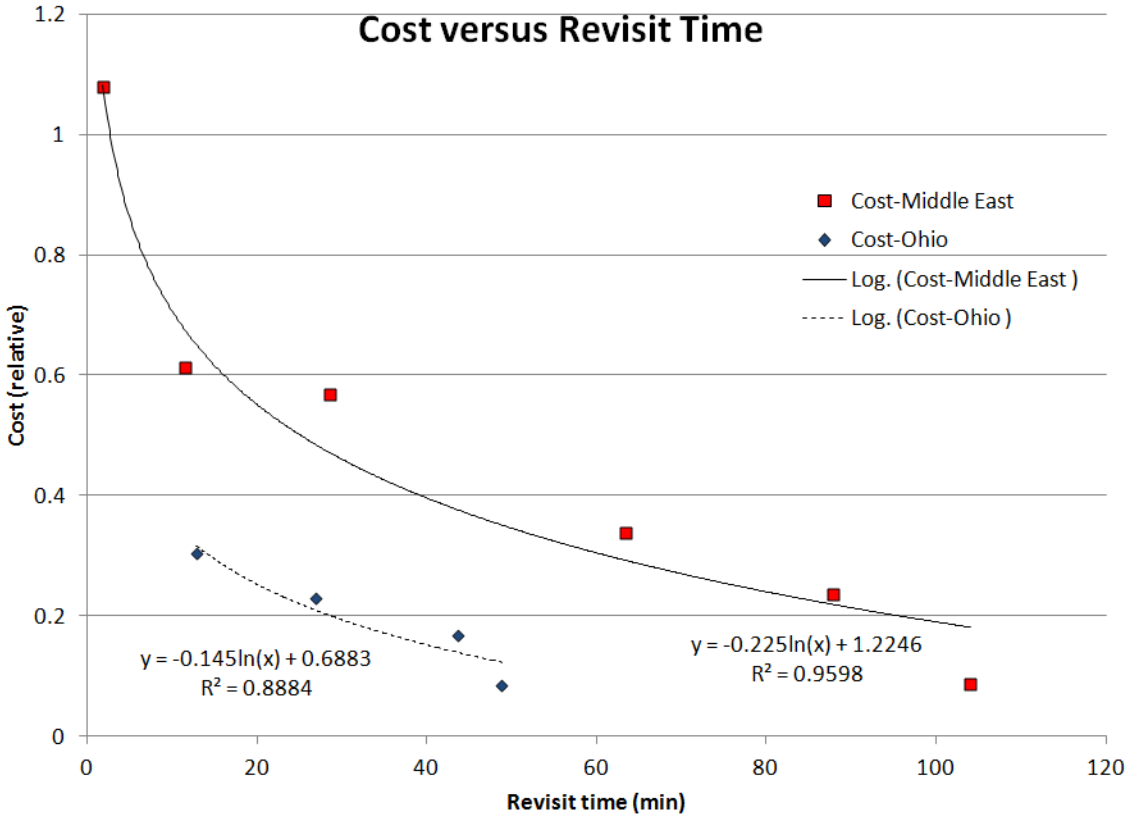


Figure 4.3: Cost versus Revisit Time with refined solutions. Again the cost is a normalized value.

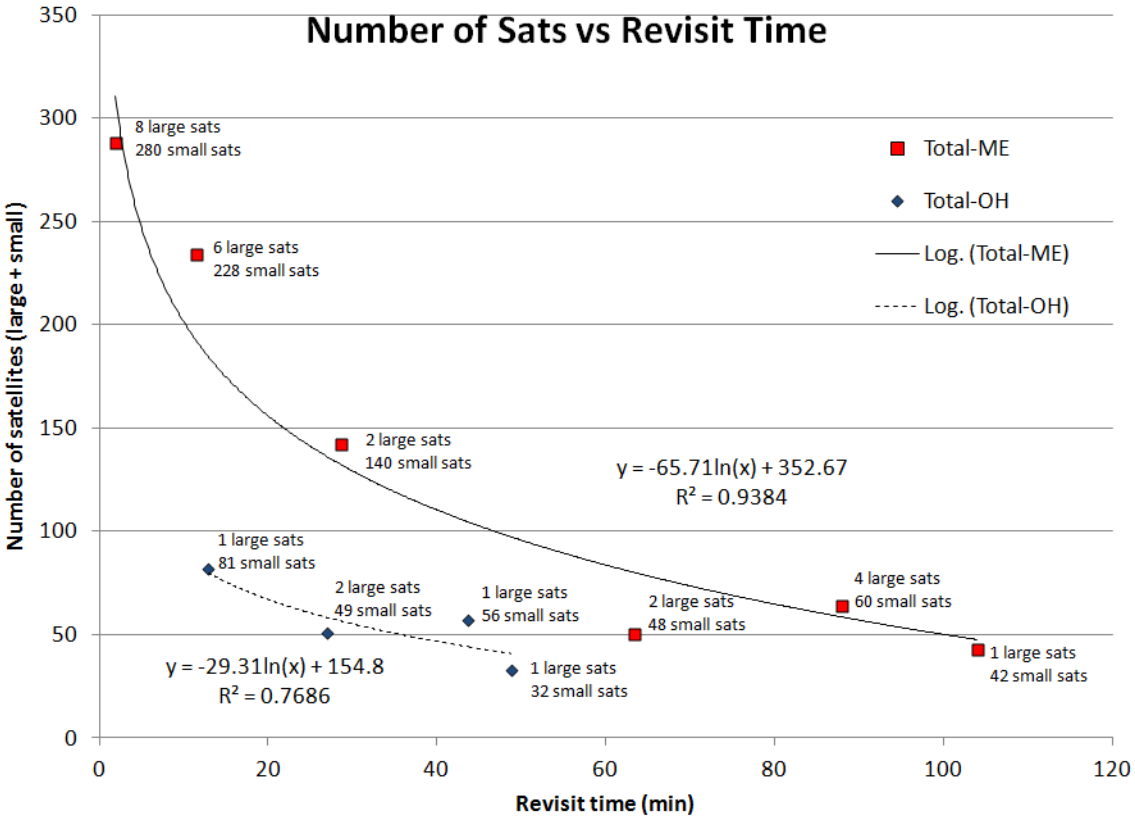


Figure 4.4: Number of Satellites versus Revisit Time with refined solutions

scales all NIIRS levels higher than they should be, but the relative NIIRS values should hold true, and the margin between our minimum NIIRS requirement (level 3) and the NIIRS values of the solutions is large.

Some interesting trends are already becoming apparent. A logarithmic relationship seems to exist between number of satellites and revisit time, as well as between cost and revisit time. Since the values are directly proportional, it is intuitive for the cost curve to have the same shape as the number of satellites curve. The number of satellites increases exponentially with lower revisit times because of the multi-dimensional nature of the problem and the wide circular footprints of each spacecraft augmented by targeted sensors. The area of the footprint is a function of the square of the radius, where the radius is proportional to the altitude. Very short revisit times will require nearly persistent coverage, for which many satellites are needed. For each minute that access is not required, that many fewer satellites are needed to provide coverage. Our disregard to launch costs for this study would not affect that relationship too much, since launch costs are one time at the beginning of life and would simply shift the curve up the cost axis. As more data points are added, the trend can be held with higher confidence. We also see that most of the solutions make use of mixed disaggregated constellations, augmenting constellations of small satellites with a few larger satellites.

The results of the GA came out seemingly at random, but ideally thousands of runs should be done to fill in all the gaps in information. However, the results seen here allow for some theories to start to be developed while more data points are gathered. For example, Figure 4.4 makes it clear that the higher latitude target, Ohio, needs fewer satellites for similar revisit times. It is notable that Ohio also has a smaller area than the Middle East target, which will affect comparisons to some degree.

Computation time for the simulation is on the order of 12 hours. The GA settles on some particular "minimum" function values over multiple runs, from which

we must take the lowest. Remember that the GA is not guaranteed to find the global minimum every time, and may get stuck in a local minimum.

4.3 Trade-offs

Cost and resolution must be balanced. Higher resolution is possible with lower altitude and larger aperture diameter. One would guess that lower altitude satellites have a shorter period resulting in faster revisit times, so higher altitude solutions should require more satellites. On the other hand, lower altitude satellites have a smaller footprint and must be augmented by more satellites. Larger diameter means bigger, more expensive satellites. The GA's contention is to find the sweet spot that gives the lowest cost, which may mean a non-intuitive solution because of the complex trade-offs found in this problem.

In light of these trade-offs, a few more charts are provided based on the simulation results in Figures 4.5, 4.6, and 4.7.

4.4 Realistic Results

The results described in the above sections include constellation designs with high altitude orbits and low sensor aperture diameters. This leads to the question of feasibility: at thousands of kilometers, with such a small optic, even low level NIIRS values may not be attainable. The issue was with the f-number, which was briefly described in Chapter 2 as the ratio of the focal length and the aperture diameter. With the assumed focal length of 10 meters used to get these results, and sensor diameters on the order of 0.1 meters, the f-number comes out to about 100. Recall from Chapter 2 that Landsat 7 has an f-number of 6 [47], so even a high end optic would not realistically be able to achieve an f-number of 100. To add realism, future implementations of this problem should use a focal length of six times the diameter.

Despite this lapse in realism, results are still accurate relative to each other, and the algorithm can be used to optimize any problem in conjunction with STK

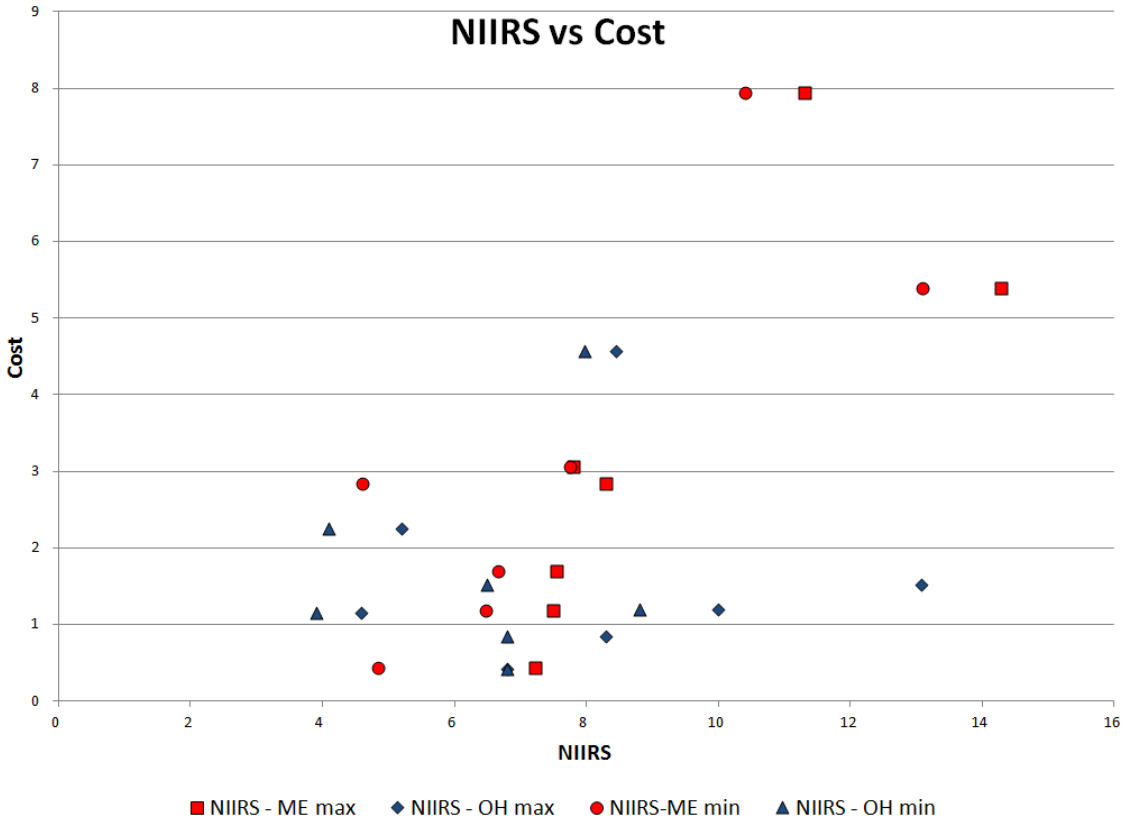


Figure 4.5: NIIRS versus Cost. This is the main trade-off of resolution versus cost. The minimum NIIRS value was used to generate this data. A general trend can be seen of increasing cost with increasing resolution, as one would expect.

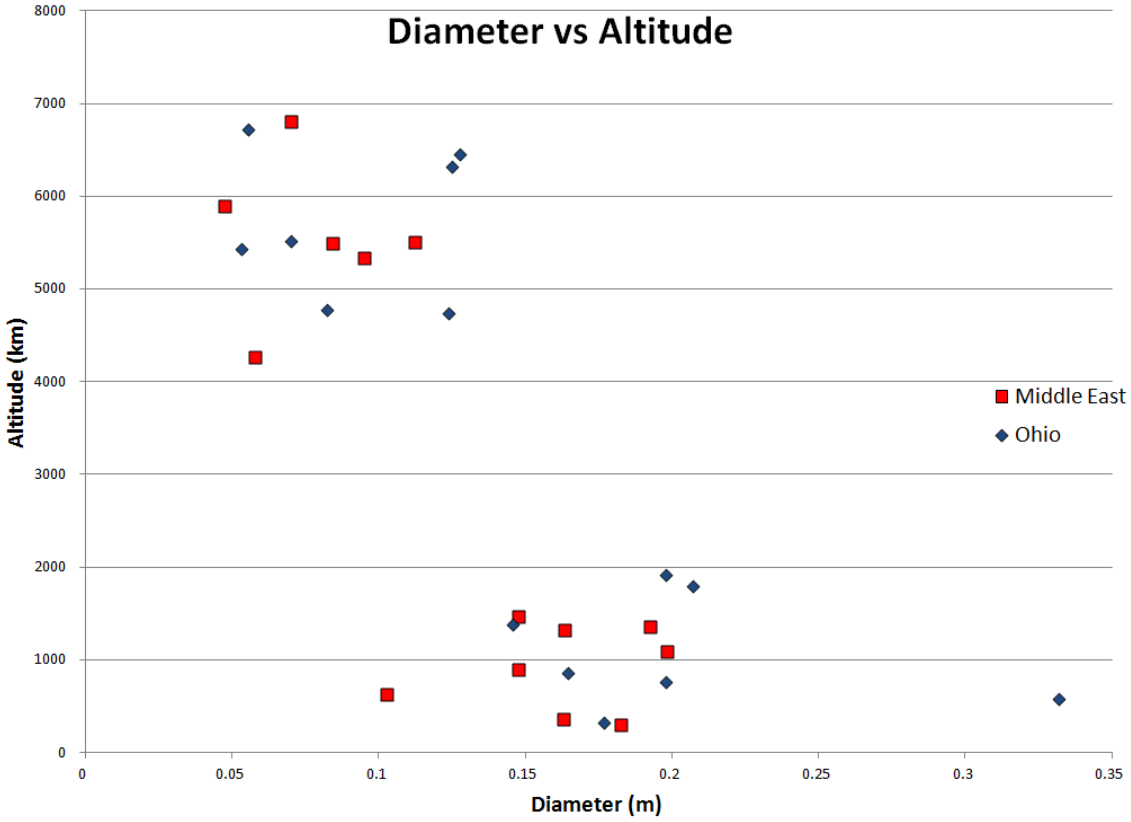


Figure 4.6: Diameter versus Altitude. The smaller satellites with the smaller aperture diameters are generally placed in higher altitude orbits and are greater in number, whereas the larger satellites are found to be more useful at lower altitudes. This is an interesting correlation to find, since one would expect the smaller aperture to be compensated for by placement in lower altitudes. It is likely that the lower altitude satellites (which are generally larger based on Figure 4.7) are used to get good resolution, while the higher satellites were primarily filling in coverage.

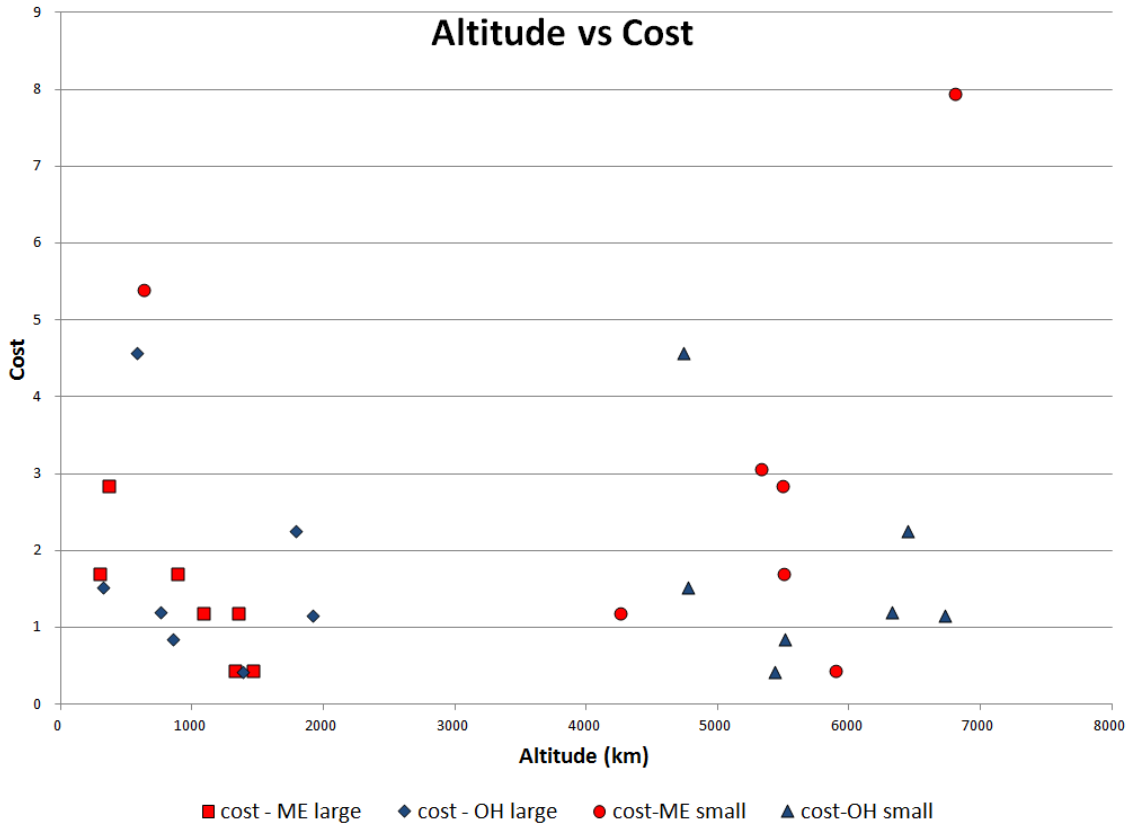


Figure 4.7: Altitude versus Cost. Here Middle Eastern target deck solutions are shown in red and Ohio target deck solutions shown in blue, with different shapes for the smaller and larger satellites. This chart shows the larger satellites clustered at lower altitudes and the smaller satellites clustered at higher altitudes. A very slight trend might be seen with higher costs at lower altitudes, but no real correlation can be drawn from this data.

propagation as long as the user has mathematical models and inputs to represent the problem.

4.5 Analysis

Many solutions found were at very high altitudes, and with a NIIRS requirement of only three, this is not a problem. For example, the 30-minute constrained Ohio case has satellites at 2000 and nearly 7000 kilometers, but gets NIIRS values of only 4. However, despite the NIIRS constraint of 3, many solutions' NIIRS values were much higher. The solutions with higher NIIRS values were the lowest cost solutions despite the lenient NIIRS requirement, which may imply that lower cost solutions are possible (with lower resolution). An interesting question is what quantitatively happens to the solution if the NIIRS requirement is increased. For higher NIIRS requirements, the low altitude required for resolution (or larger aperture diameter) must be balanced out with the higher altitudes needed for coverage. Many solutions used altitudes in the hundreds of kilometers to yield much higher NIIRS values. A single run with a NIIRS constraint of 8 (see the last line for the Middle East target deck in Table 4.1) yielded a much higher cost than comparable solutions. The trend between NIIRS requirements and costs should be further investigated.

The solutions shown in Figure 4.6 are counterintuitive, and may be a result of a flaw in the algorithm. The algorithm may be led to put smaller satellites at higher altitudes because the footprint is wider at higher altitudes and provides better coverage, whereas the larger satellites have a large footprint even at lower altitudes. The high f-number used likely allowed this trend, as the small satellites at very high altitudes could still meet the NIIRS level three constraint. In actuality, the NIIRS level reached by small apertures at high altitudes would be much smaller than three. The user should change the focal length as described above to yield more realistic results.

We see that higher latitudes (Ohio) require fewer satellites to get similar revisit times. This may be intuitive if one considers that higher inclinations see higher

latitudes more frequently. Equatorial targets may be more difficult to cover. This also may be partially attributed to the difference in area between the Middle East target and the Ohio target.

Another interesting point is that although solutions had the option of using no small or large satellites, the solutions still used a few more expensive large satellites. In this way, the genetic algorithm provides evidence that mixed constellations are more cost-effective than constellations of a single type (size) of satellite for providing target coverage.

Remember that the cost model values are relative, not real, and so actual costs are likely to be higher than seen here. Also, all solutions found above about 1000 km will need radiation hardening due to the Van Allen radiation belt.

4.6 Recommendations

The cheapest solution found so far was a 104 minute response time with 43 satellites for the Middle East, costing a relative value of 0.09, and a 48.8 minute response time with 33 satellites for Ohio, costing a relative value of 0.08. The fastest solution found was 1.89 minutes with 288 satellites for the Middle East, costing a relative value of 1, and 12.9 minutes with 82 satellites for Ohio, costing a relative value of 0.3. The commander of the mission would need to make the call on which solution to use to make best use of resources and time, and the commander's choice may be a compromise between response time and cost. The average cost per satellite also must be weighed against the total cost of the covering constellation (see Table 4.2). The user must consider the timeline and budget of the project when weighing between constellation and per satellite costs.

The GA was able to get response times within 15 minutes for a relative cost of 0.3, which might be good enough for the Army's mission (remember from Chapter 2 that they hope for approximately 10 minutes response time); more runs might yield even more cost-effective results. The solution with a response time within only two

ME - Cost	Max revisit time
1	2 minutes
0.6	11.5 minutes
0.09	104 minutes

OH - Cost	Max revisit time
0.3	13 minutes
0.08	50 minutes

Table 4.3: Cheapest and fastest results for the Middle East and Ohio target decks, normalized to $5 billion referenced to the cost model.

minutes is certainly the best for responsiveness, but the high pricetag might make this solution unrealistic.

4.7 Comparison with Other Work

Captain Steven Ingraham's thesis [29] used maneuverable satellites. Combining his work with this GA algorithm could yield interesting results, since this thesis assumed a fixed constellation. Maneuverability could allow more flexibility in resolution and cost.

Ingraham's thesis determined that the inclination to use for optimal coverage of a target area is the sum of the target latitude and the Earth central angle of the spacecraft [29]. Using calculations provided in his thesis (see Chapter 2) and assuming a minimum elevation angle of 5 degrees, the optimal inclination to use for our targets can be seen in Table 4.4.

Some of the inclinations found by the GA are as close as within ten degrees to the calculations provided by Capt. Ingraham, though many are very far off. The GA uses a combination of small and large satellites that can be placed in different orbits. A different optimal inclination might be found if Capt. Ingraham's work accounted for mixed orbits rather than a single constellation. The deterministic process used

Middle East			Ohio		
alt (km)	incl (deg)	opt incl	alt (km)	incl (deg)	opt incl
1462.7	31.4	71	576.7	49.4	58
5898.1	88.5	96	4741.3	72.7	90
1326.9	33	69	759.3	46.8	62
4259.2	77.3	88	6324.4	48.8	95
1353.8	54	70	1384.3	41.2	70
5500.7	58.9	92	5435.3	52.9	92
1085.9	78	66	854.8	65.2	63.5
5490.8	52	92	5512.1	54.4	92
892.7	84.5	64	1920.1	48.7	75
5330.9	38.9	92	6725.6	38.4	96
295.2	41.9	52	326.5	49.6	53
628.5	50.5	59	4777.2	44	90

Table 4.4: Optimal inclinations versus inclinations calculated by the algorithm. The altitudes and inclinations are taken right out of Table 4.1 and the "optimal" inclinations are calculated using Capt. Ingraham's equations [29]. The inclinations that are close to Capt. Ingraham's optimal inclination (arbitrarily defined as within 10 degrees) are generally lower in altitude than the farther off ones.

by Capt. Ingraham is much different than the stochastic process used by the GA for finding optimal inclinations.

Using the same cost models as this research, TACSAT's constellation yields a cost of $990 million dollars for 10 satellites (52 minute revisit), and $1.7 billion dollars for 20 satellites (22 minute revisit) [35]. Running each through the sensor performance code, both have a NIIRS value of 3.2 over the Middle Eastern theater. Comparing the TACSAT numbers to the results found in this thesis, and keeping in mind the purely relative nature of the cost model used, a 104 minute revisit is found for $430 million, and a $1.7 billion dollar constellation only yields a revisit of 66 minutes, but the NIIRS values for these are about seven. The higher cost could be attributed to the higher NIIRS values rendered by the solution.

Another important difference is that these works did not account for constellation costs, which affect the size and number of spacecraft that can be used. While Capt. Ingraham and TACSAT optimize for more efficient constellations (maximum coverage), this thesis attempts to balance cost with response time of solutions (minimizing cost).

4.8 Summary

In this chapter, the simulation results were given and a trend could begin to be established between revisit time and cost, as well as revisit time and constellation size. The last chapter will provide conclusions of this thesis.

V. Conclusions

The main contribution of this work was the genetic algorithm model that could be run in Matlab, in conjuction with STK, to optimize constellations for desired parameters, and especially the automation that the code offers in providing solutions for any given target deck. The results showed that augmenting a constellation of large satellites could be effective in increasing capabilities, and that a stand-alone constellation of small, inexpensive satellites could accomplish the mission of nearly persistent coverage at a minimum NIIRS level of three with just a few exquisite (large) satellites to augment it. This work should be referenced by sponsors to save money and increase capabilities as space architectures are made more resilient and efficient.

5.1 Challenges

The most challenging part of this research is the limitation brought about by the computation time. Each time a change is made, the user must wait overnight to see its effects. This issue is a resource limitation; more available computing power would increase productivity of this algorithm.

Another challenge was accomplishing STK tasks within Matlab code. To execute STK tasks in Matlab, multiple references are often needed, and it is necessary to splice example Matlab code with example commands from STK. The user must realize that the STK name and Matlab name for a particular object are not necessarily the same. STK provides an object model, which maps every possible object and its children (sublayers) within STK, but adapting these objects into a Matlab-friendly environment is not a trivial task.

5.2 Recommendations for Future Work

The benefits of disaggregation are clear, but the biggest obstacle is its characteristic of low reliability. To overcome this, disaggregation should be applied to existing missions and observations of performance recorded; many data points are required and the easiest way to get them is to conglomerate results of various missions that

utilize disaggregated architectures. Well-established companies with existing constellations could experiment with disaggregation concepts, and by pooling the resulting data, no single company would have to take too large a risk.

To that end, Eric Fanning, undersecretary of the Air Force, was quoted this year in a March 5 briefing with reporters, saying, "The Air Force is committed to disaggregation [55]." In the same briefing, Major General Robert McMurry, director of space programs in the Air Force acquisition office, said the Advanced Extremely High Frequency (AEHF) system will be the first existing program to incorporate disaggregation [55]. Following this trend, disaggregation concepts should be inserted into relevant, existing programs to reap the potential benefits.

The code that resulted from this work could also be adapted for various other uses, by changing the design vector, constraint functions, and cost function to suit the user's unique needs.

The model created in this research could be expanded and further explored. Many interesting questions were not able to be investigated that were mentioned in the analysis in Section 4.5. More future work to be done includes:

- Model launch vehicle costs
- Model ground architecture
- Multi-objective architecture optimization
- Prioritization of targets (other than STK default scheduling algorithm)
- Investigation of whether different optimizers change solutions
- Simultaneous access calculations
- Runs of the algorithm with more or different target decks and NIIRS levels

Launch vehicle costs were not considered in this thesis. A simple, linear launch cost model would be an easy addition to future iterations of this work. Further research should investigate how the cost function might change by using different launch vehicles or bundling spacecraft together on launch vehicles in various configurations.

This thesis has not accounted for downlinking data and images to a groundstation or handheld device of any sort. This should be accounted for and modeled in future renditions of this research.

This thesis has focused on single-objective architecture optimization; there are multi-objective optimization techniques available that should be applied to disaggregation concepts. This would include optimizing the constellation with respect to not only cost, but also response time and number of spacecraft in the constellation. Lower cost, faster response time, and fewer spacecraft are better. This research also concentrated on maximum revisit time as a figure of merit; mean, minimum, and maximum coverage gaps, as well as other figures of merit, could be used in a multi-objective optimization.

This thesis has not attempted to assign weights to targets. STK has a scheduling algorithm that it automatically applies to sensing spacecraft. Future research should assume that some user-requested targets will be higher priority than others, and assign weights to targets accordingly. This concept will only be viable for specific examples of target decks, and in the real world will need to be evaluated for actual targets.

This thesis used exclusively Matlab's genetic algorithm to optimize the constellation. Future work may involve exploring different optimization algorithms. Examples of different optimizers were discussed briefly in Section 2.3. Further improvements to the model can be made within the GA as well. Using a higher population size and a lower function tolerance will yield better results.

Computers with more cores may be available to speed up the algorithm. If there is a way to calculate access for all targets simultaneously, that would speed up calculation time even more. Extracting data points without generating reports would also reduce calculation time. Within the model itself, the calculation could be done faster by further restricting the bounds. For example, a smaller range of inclinations, based on the location of the target region, would be plausible. More study should be done to determine appropriate bounds.

Analyzing the results of disaggregation will take a long time, and can only be furthered by continually applying it to different assumed target decks. The more data that is available, the more knowledge we will have of its merits and drawbacks. Even the Middle East and Ohio target decks explored here could be further expounded upon.

5.3 Conclusions

This thesis determined that the Army's mission objectives - user-tasked coverage of a target region within minutes - could be reasonably accomplished in under 12 minutes, using 234 satellites (large and small) at a NIIRS level of seven (better than the required three). A 2 minute response time is possible but may be too costly, and a 100 minute response time could be achieved but might not provide fast enough response. The smallest notional average cost per satellite is $10 million, but that includes both large and small satellites that cannot maneuver, so the Army's estimate of around $1 million per satellite may be achievable in a smaller constellation of maneuverable satellites. These solutions assume that the targets covered per pass are limited in number and located in the Middle East as described in Chapter 3.

This work did not assume maneuverable satellites, but the savings from using fewer satellites for coverage might be offset by the cost of fuel used for maneuvering. The solutions in this thesis are representative of disaggregated imaging systems in mixed constellations, and help to confirm the belief that an augmenting constellation of small satellites can add increased capability to a constellation of monolithic spacecraft. We must remember that in actuality, small spacecraft typically have less capability, and so larger spacecraft may be needed to technically accomplish the proposed mission.

The genetic algorithm can optimize this problem within 12 hours. Our method of letting the scenario propagate automatically within the genetic algorithm saved time; the population of spacecraft was much faster than it would have been to do it by hand. For the number of runs the genetic algorithm requires, the automatic

interface is best because it can be allowed to run overnight without oversight. Also, computers with more cores are available and can be used to significantly reduce the computation time.

In general, an existing constellation of large satellites used to image high-resolution targets with a lenient revisit time could be augmented by a constellation of small satellites focusing on a region of interest to provide higher resolution imagery and a much shorter revisit time.

Shifting the collective mindset from monolithic spacecraft to modular, disaggregated systems may be the gateway to having myriad advanced and amalgamated systems in orbit and beyond. As Neil Armstrong so eloquently phrased it, disaggregation could be "one small step for [a] man; one giant leap for mankind."

Appendix A. Astrodynamics

To understand what the design vector parameters represent, a few definitions are necessary, and are summarized in Table A.1. *Apogee* is the point of the orbit farthest from the Earth, while *perigee* is the point of the orbit closest to Earth, and is exactly 180° from apogee. The *vernal equinox* is the point in the sky where the Sun is located on the first day of spring [4], and is used as an inertial reference point. All angles are measured with the center of the Earth as an origin point. The *equatorial plane* is a plane extending into space from the Earth's equator.

Semi-major axis is defined as half the linear distance from apogee to perigee. For circular orbits, such as those used in this research, it is simply the distance from the center of the Earth. The altitude can be calculated by subtracting the mean radius of the Earth, 6378.137 kilometers, from the semimajor axis [4].

Eccentricity quantifies the circularity of an orbit. A circular orbit has an eccentricity of zero; the orbit becomes more elliptical as eccentricity approaches one. Higher eccentricities represent hyperbolic orbits, and are not discussed in this paper.

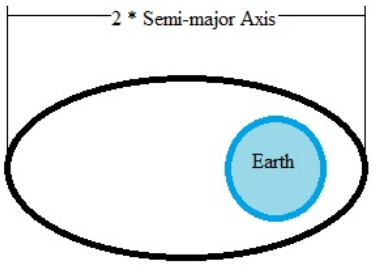


Figure A.1: Semi-major Axis

Inclination is the angle between the equatorial plane and the orbital plane, with 90° representing a polar orbit.

Right ascension of the ascending node is the angle measured eastward between the vernal equinox and the ascending node of the orbit, where the orbit crosses the equatorial plane travelling northward.

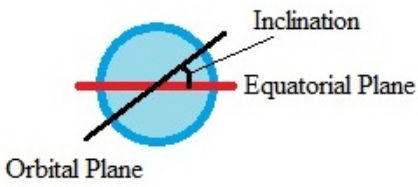


Figure A.2: Inclination

Argument of perigee is the angle measured in the direction of satellite motion between the

Vernal equinox	Inertial reference point
Semi-major axis	Radius of orbit
Inclination	Angle of orbit with respect to Earth's equator
Right ascension of the ascending node	Angle of orbit with respect to the vernal equinox
True anomaly / Mean anomaly	Spacecraft's position in orbit

Table A.1: Astrodynamics Definitions for Circular Orbits

ascending node and perigee. For circular orbits, this measurement exists but is irrelevant.

True anomaly is the angle between perigee and the spacecraft. For a circular orbit, the true anomaly is equivalent to mean anomaly [4], M, which is used in this paper. The true anomaly is the only one of the classical orbital elements that changes over time, since it describes the spacecraft's position in space as it travels around the Earth.

Appendix B. NIIRS Parameter Derivations

First the slant range is determined for each target by STK, and used to compute the ground sample distance (GSD):

$$GSD = R_{slant}\frac{d/f}{\sqrt{\cos\psi}}$$

where R_{slant} is slant range determined by STK, d is the width of the focal plane detector in meters assumed to be $30 * 10^{-6}$ to match Teledyne Imaging Sensors CHROMA Visible-Infrared Focal Plane Array, f is the focal length in meters assumed to be 10 to match Kestrel Eye [31], and ψ is elevation angle of the spacecraft computed using the arccosine of the semi-major axis divided by the slant range. The ground sample distance (GSD) of the Kestrel Eye program, 1.5 meters [31], was used to verify the accuracy of this equation by assuming Kestrel Eye data from the Army SBIR [31]: 500 km range, $30 * 10^{-6}$ m pixel width, 10 m focal length. Also assuming a nadir pointing sensor ($\psi = 0$) gives

$$GSD = \frac{500000\frac{30*10^{-6}}{10}}{1} = 1.5m$$

The relative edge response (RER) is computed next using a modulation transfer function (MTF). For this equation, we assume no jitter, smear, or wavefront aberration. We also assume a circular aperture with no obscurations. This simplest case includes only the most significant factors: the sampling of the detector/telescope combination, and the diffraction characteristics of the telescope. The system MTF is given by

$$MTF_{system} = MTF_{sampling} * MTF_{diffraction}$$

where both MTFs are normalized over the same frequency. The sampling and diffraction MTFs are

$$\begin{aligned} X_{sampling} &= \frac{f}{d} \\ MTF_{sampling} &= \frac{\sin(\pi\chi)}{\pi\chi} \\ X_{diffraction} &= \frac{D}{\lambda} \\ q &= \frac{X_{sampling}}{X_{diffraction}} \\ MTF_{diffraction} &= \frac{2}{\pi}(\arccos(\chi/q) - \chi/q\sqrt{1-(\frac{\chi}{q})^2}) \end{aligned}$$

where χ is the cutoff frequency, D is the aperture diameter (a design variable), and λ is the mean wavelength in meters, assumed to be $0.65 * 10^{-6}$ for visible light. The edge response (ER) defines the contrast of a picture, and is defined by

$$ER(\zeta) = 0.5 + \frac{1}{\pi}\int_0^{1/q}(\frac{MTF(\chi)}{\chi}\sin(2\pi\chi\zeta))\,\mathrm{d}\chi$$

where ζ is the number of pixels offset measured from the edge. The MTF here is the system MTF including all contributions (in our case, only sampling and diffraction). The RER and overshoot parameter (H) can then be found.

$$\begin{aligned} RER &= ER(0.5) - ER(-0.5) \\ &= 2\int_0^{1/q}(MTF(\chi)\frac{\sin(\pi\chi)}{\pi * \chi})\,\mathrm{d}\chi \end{aligned}$$

H is the peak value of the ER over the range $\zeta = 1$ to 3, or the value of the ER at $\zeta = 1.25$ if the ER monotonically increases over that range. This is done in the Matlab code using if statements and using a search fuction to find the maximum value of the ER evaluated at intervals of 0.01 between 1 and 3.

For G, [54] suggests using half of the signal-to-noise ratio (SNR) as a conservative estimate. However, this yielded values too high for G. Thurman [56] summarizes the

	Minimum	Mean	Maximum
GSD	3 in	20.6 in	80 in
RER	0.2	0.92	1.3
G	1	10.66	19
SNR	2	52.3	130
G/SNR	0.01	-	1.8
H	0.9	1.31	1.9

Table B.1: Ranges for GIQE parameters [56]

ranges for GIQE values, shown in Table B.1. The mean value for G, 10.66, was used in all cases as a rough approximation. The rest of the ranges in the table were used to verify all sensor performance values.

The SNR is the most complicated calculation with varying parameters, but typical values are used to determine a rough approximation.

$$SNR = \frac{\Delta\rho * R_S}{\sqrt{\rho * R_S + R_H}} * \sqrt{K * t_{int}}$$

where target reflectance must be assumed. The GIQE Users Guide recommends using a delta reflectance $\Delta\rho$ of 0.08 for computing the target signal, and a reflectance ρ of 0.15 to compute the random noise due to the target signal. A conservative estimate for integration time, t_{int}, is 0.001 seconds. K for visible near infra-red (VNIR) is $1.63 * 10^{-13} electrons * photons^{-1} * m^2 * sr$, and scene and haze spectral radiances (R_S and R_H, respectively) are taken as the rough area under the curve, over wavelength of interest 0.4 to 1 micron, in the graph shown in Figure B.1. For scene radiance, the area under R_{S^λ} is approximately

$$R_S = 0.5 * (0.9 - 0.4) * (1.35 * 10^{21} - 6 * 10^{20}) + (0.9 - 0.4) * 6 * 10^{20}$$

and the area for haze reflectance, under R_{H^λ} is approximately

$$R_H = 0.5 * (0.9 - 0.4) * 2 * 10^{20}$$

These approximations assume that the Sun and sensor are both directly overhead from the target, and so actual numbers will differ.

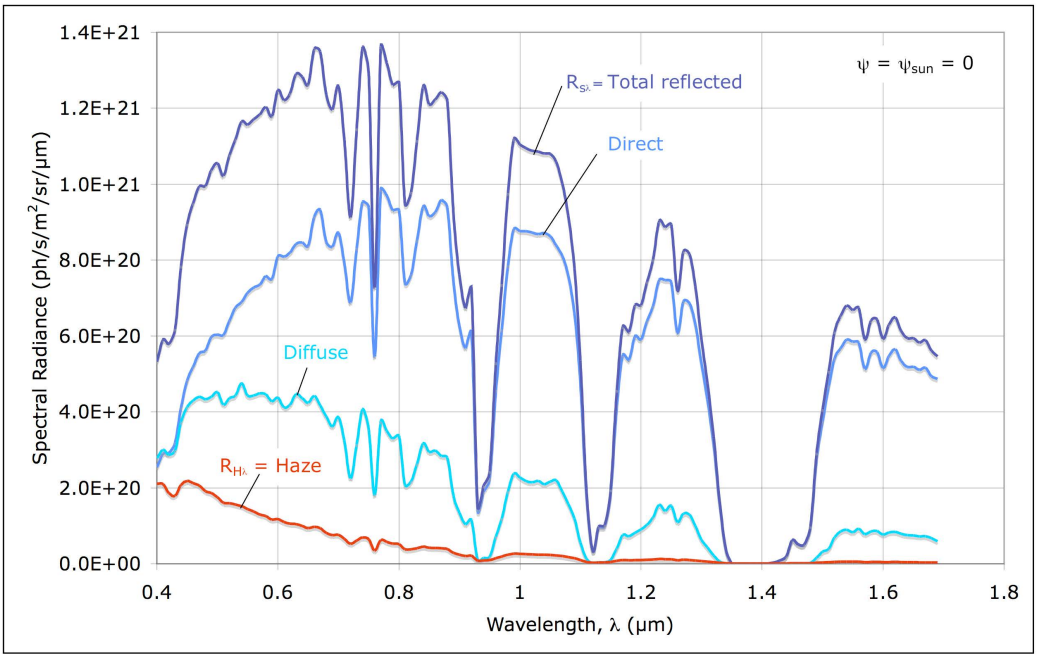


Figure B.1: Reflected and Haze Spectral Radiances (Top of Atmosphere) [54]. The reflected (or scene) and haze radiances needed for SNR calculations can be computed as the area under the curve over the wavelength range of interest.

The coefficients c_0 through c_4 are assumed to be the GIQE Version 4 (GIQE 4) coefficients in metric units for $RER < 0.9$, found in Table 10 of Auelmann's article [54]:

$$c_0 = 5.205$$

$$c_1 = -3.16$$

$$c_2 = 2.817$$

$$c_3 = -0.656$$

$$c_4 = -0.344$$

Appendix C. Reference Solution Cost Calculation

This Appendix will go through an example calculation for the following design vector:

$$x = [1 \quad 1212.1 \quad 217.11384.3 \quad 041.2 \quad 175.50 \quad 200.1456$$
$$8 \quad 4115.8 \quad 76.45435.3 \quad 052.9 \quad 212.10 \quad 200.053]$$

This design vector yields a 48.8 minute revisit time and costs $420 million dollars. The cost is calculated using a typical imaging spectoral payload from [4], which has the characteristics specified in Table C.1.

The sizing ratio of the sensor compared with the reference sensor is calculated using the diameters of each:

$$R = \frac{A_i}{A_o}$$
$$= \frac{0.1456}{0.18} = 0.8089$$

for the large spacecraft and

$$R = \frac{0.053}{0.18} = 0.2943$$

Parameter	Symbol	Value
Aperture diameter	A_o	0.18 m
Mass	W_o	250 kg
Power	P_o	225 W
Data rate	DR	11 Mbps

Table C.1: Reference Payload Characteristics [4]

for the small spacecraft. For $R < 0.5$ (the small spacecraft), K will be 2. Otherwise K is 1.

$$\begin{aligned} W_i &= K * R^3 * W_o \\ &= 0.8089^3 * 250 = 132.3187 \\ P_i &= K * R^3 * P_o \\ &= 0.8089^3 * 225 = 119.0868 \end{aligned}$$

for the large and

$$\begin{aligned} W_i &= 2 * 0.2944^3 * 250 = 12.7425 \\ P_i &= 2 * 0.2944^3 * 225 = 11.4683 \end{aligned}$$

for the small, so the dry mass of each spacecraft is

$$\begin{aligned} m &= W_i/0.31 - W_i \\ &= 132.3197/0.31 - 132.3197 = 294.5158kg \end{aligned}$$

for the large and

$$m = 12.758/0.31 - 12.758 = 28.3624kg$$

for the small. The data rate is assumed to be the same as the reference payload. So the NICM model payload cost is:

$$NICM = 1163 * W_i^{0.426} * P_i^{0.414} * (DR)^{0.375}$$

which is 165,700 for the large and 23,228 for the small.

Next, the USCM NRE cost is calculated as the sum of the spacecraft, payload, integration and test (IAT), program level, and ground equipment costs, and is only valid for the large spacecraft:

$$
\begin{aligned}
NRE_{spacecraft} &= 110.2 * m \\
NRE_{payload} &= 0.6 * NICM \\
NRE_{IAT} &= 0.195(NICM + USCM_{spacecraft}) \\
NRE_{program} &= 0.414(USCM_{IAT} + USCM_{spacecraft}) \\
NRE_{AGE} &= 0.421(USCM_{spacecraft})^{0.907} * 2.244
\end{aligned}
$$

Plugging in the appropriate values and summing these together, the USCM cost for the large spacecraft 193,340. The TFU cost is calculated next as the sum of payload, spacecraft, IAT, program, and launch and operational support costs:

$$
\begin{aligned}
TFU_{payload} &= 0.4 * NICM \\
TFU_{spacecraft} &= 289.5 * m^{0.716} \\
TFU_{IAT} &= 0.124(TFU_{spacecraft} + TFU_{payload} \\
TFU_{program} &= 0.320(TFU_{payload} + TFU_{spacecraft} + TFU_{IAT}) \\
TFU_{LOOS} &= 5850
\end{aligned}
$$

which equates to 129,360.

Finally, the small spacecraft cost is calculated using the SSCM model, as follows. First the non-recurring engineering cost:

$$\begin{aligned} SSCM_{spacecraft} &= 1064 + 35.5m^{1.261} \\ SSCM_{payload} &= 0.4 * SSCM_{spacecraft} \\ SSCM_{IAT} &= 0.139 * SSCM_{spacecraft} \\ SSCM_{program} &= 0.229 * SSCM_{spacecraft} \\ SSCM_{GSE} &= 0.066 * SSCM_{spacecraft} \\ SSCM_{LOOS} &= 0.061 * SSCM_{spacecraft} \end{aligned}$$

which gives a value of 3545.9. The TFU costs are

$$\begin{aligned} &0.4 * SSCM_{spacecraft} + 0.4 * SSCM_{payload} + 1 * SSCM_{IAT} \\ &+0.5 * SSCM_{program} + 0 * SSCM_{GSE} + 1 * SSCM_{LOOS} \end{aligned}$$

for 3038.6. The total cost is the sum of all of these final values.

$$1 * (NRE + TFU) + 32 * SSCM = 423,480$$

This is calculated in thousands of dollars, so the actual price of the constellation will be $423 million.

D.1 Genetic Algorithm Code

Listing D.1: Appendix1/targets.m

```
Scen.Title      = 'Imager';
Scen.TimeStep   = 5;
Scen.StartTime  = '16 Sep 2013 16:00:00.000';
Scen.EndTime    = '18 Sep 2013 16:00:00.000'; % 48 hour scenario
[uiapp, root]       = STK.Initialize(Scen);

% for n=1:20
%     target(n)=STK.create_target(horzcat('target',num2str(n)),...
   root,[randi([-45,75],1),randi([-180,180],1),0]);
% end

% Middle East
for n=1:5
    target(n)=STK.create_target(horzcat('target',num2str(n)),root...
        ,[36,41,0]); %upper left
end
for n=1:5
    target(n)=STK.create_target(horzcat('target',num2str(n+5)),...
        root,[36,46,0]); %upper right
end
for n=1:5
    target(n)=STK.create_target(horzcat('target',num2str(n+10)),...
        root,[30,42,0]); %lower left
end
for n=1:5
    target(n)=STK.create_target(horzcat('target',num2str(n+15)),...
        root,[30,48,0]); %lower right
end
for n=1:5
    target(n)=STK.create_target(horzcat('target',num2str(n+20)),...
        root,[33,44,0]); %center (near Baghdad)
end
```

```
% % Ohio
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n)),...
    root,[42,-85,0]); %upper left
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+5)),...
    root,[42,-80.5,0]); %upper right
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+10)),...
    root,[39,-85,0]); %lower left
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+15)),...
    root,[39,-81,0]); %lower right
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+20)),...
    root,[40,-83,0]); %center (Columbus)
% end

% % N. Korea
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n)),...
    root,[41,125,0]); %upper left
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+5)),...
    root,[43,130,0]); %upper right
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+10)),...
    root,[38,125,0]); %lower left
```

```
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+15)),...
    root,[38.5,128,0]); %lower right
% end
% for n=1:5
%     target(n)=STK.create_target(horzcat('target',num2str(n+20)),...
    root,[39,126,0]); %center (Pyongyang)
% end

root.SaveScenarioAs('I:\My Documents\THESIS\STK_Matlab\Imager\...
    Matlab\180minute\TargetDeck.sc');
```

Listing D.2: Appendix1/image_ga.m

```
% image_ga  ->  Disaggregated Space System Optimization Routine ...
    using the
% matlab genetic algorithm routine
clear all;
close all;
clc
% for trial=1:10

    %GA setup
    % Initialize global variables (sensor angles)
    global nvars ntargets

    %**************************USER INPUTS...
       ***********************************
    % Lower and Upper Bounds on System Type Parameters
    % Type: Unmanned Spacecraft (large)
    vec(1).num=[0 0 0 0;10 10 360 360]; % num planes, sats/plane, ...
       true an phasing; RAAN increment
    vec(1).location=[200 1e-6    30    1e-6  1e-6    20; 2000 1e-6...
        90 360 1e-6 20]; % alt e i om w M
    vec(1).design=[.1445;.3774]; % diam
```

```
% Small Spacecraft (small)
vec(2).num=[0 0 0 0;10 10 360 360]; % num planes, sats/plane, ...
    true an phasing; RAAN increment
vec(2).location=[200 1e-6    30    1e-6  1e-6    20; 7000 1e-6...
     90 360 1e-6 20]; % alt e i om w M
vec(2).design=[.0471;.1612]; % diam

% How many parameters per system type?
ntargets=25; % RUN TARGETS.m!!
nvars = 11;    % Number of variables
types=numel(vec);

%**********************END USER INPUTS...
    ***********************************


ObjectiveFunction = @cost;
for n=1:types
    LB(nvars*n-nvars+1:nvars*n) = [vec(n).num(1,:) vec(n)....
        location(1,:) vec(n).design(1,:)]
    UB(nvars*n-nvars+1:nvars*n) = [vec(n).num(2,:) vec(n)....
        location(2,:) vec(n).design(2,:)]
    ints(2*n-1:2*n) = [nvars*n-nvars+1,nvars*n-nvars+2];% num ...
        planes and num sats/plane must be integers for each ...
        type
end
ConstraintFunction = @constraint;
options = gaoptimset('PlotFcns',{@gaplotbestf,@gaplotbestindiv...
    },'StallGenLimit',10, ...
    'Display','iter','OutputFcns',@gaoutputfcn,'PopulationSize...
        ',20,'TolFun',10,'Generations',50);

%GA solver routine
try
```

```
43          [x,fval,exitflag,output, population, scores] = ga(...
               ObjectiveFunction,nvars*types,[],[],[],[],LB,UB, ...
                ConstraintFunction,ints,options)
            modifier='success';
        catch me
            modifier='failed';
48          report=getReport(me);
        end

        filename= strcat(date,modifier);
        save(filename)
53      saveas(gcf,filename)

    % end
```

Listing D.3: Appendix1/cost.m

```
 function Cost = cost(x)
 global nvars

%x is defined as a point solution to test the function prior to ...
   optimizing
 5 %x=[10.0000    9.0000    0.2000  200.0000    0.0500  200.0000]

% x = [ nplanes  nspp       truan    a    e     i   RAAN omega M ...
       dia ]
%  x = [ 3        1          180     200  1e-6  98.6   0    0   20...
       1.2...
%        7        1          180     200  1e-6  98.6   0    0   20...
       .05];
10 % nvars=10;

% % Change design vec to variables
% %*****************REFERENCES...
   ********************************************
%Characteristics of typical imaging spectoral payload - pg 510
```

```
Ao=.18; %(Initial ref sensor aperature diameter)
Wo=250; %Weight in kg of reference payloads
Po=225; % Ref power, W
dro=11; % Ref data rate, Mbps
% ...
    %*************************************************************************...

types=numel(x)/nvars;
for n=1:types % n refers to sat type
    nsats(n)=x(n*nvars-nvars+1)*x(n*nvars-nvars+2);
    diam(n)=x(n*nvars);

    if nsats(n)>0 z(n)=1; else z(n)=0; end

    %system sizing summary
    Ai(n)=diam(n);  %(New instrument aperture diameter) - sec ...
       17.2.6 pg 516

    R(n)=Ai(n)/Ao;  %Sizing Ratio
    if R(n)<.5; K(n)=2; else K(n)=1; end
    Wi(n)=K(n)*R(n)^3*Wo; % design sensor mass
    Pi(n)=K(n)*R(n)^3*Po; % design sensor power
    Spacecraftdrymass(n)=(Wi(n)/.31)-Wi(n); %Table 14-18 pg 422, ...
       LEO s/c w/ prop **need to subtract p/l mass, right?

    % Calculate NICM NRE+TFU for Optical Earth-Orbiting Payload (...
       pg 303 11-14)
    %   (Historically 60% NRE, 40% Recurring [Old SMAD pg 799])
    %   SMAD pg 297: Some risk associated with combining models; ...
       combined b/c
    %   optical payloads omitted from USCM8 model
    payloadcost_NICM(n)=1163*Wi(n)^.426*Pi(n)^.414*dro^.375;
end
```

```
%Calculate USCM NRE cost -> non-comm sat, SME-SMAD table 11-8 pg ...
    298
Spacecraftcost_USCM=110.2*Spacecraftdrymass(1);
payloadcost_USCM=0.6*payloadcost_NICM(1); % 60% NICM is NRE
IAT_USCM=.195*(payloadcost_USCM+Spacecraftcost_USCM);
ProgramLevel_USCM=.414*(IAT_USCM+Spacecraftcost_USCM);
AGE_USCM=.421*(Spacecraftcost_USCM)^.907*2.244;

TotalCost_USCM_NRE=payloadcost_USCM+Spacecraftcost_USCM+IAT_USCM+...
    ProgramLevel_USCM+AGE_USCM;
C(1)=TotalCost_USCM_NRE;

%Calculate USCM Theoretical First UNIT (TFU) cost, SME-SMAD pg 299...
    table 9
payloadcost_USCM_TFU=0.4*payloadcost_NICM(1); % 40% NICM recurring
Spacecraftcost_USCM_TFU=289.5*Spacecraftdrymass(1)^.716;
IAT_USCM_TFU=.124*(Spacecraftcost_USCM_TFU+payloadcost_USCM_TFU);
ProgramLevel_USCM_TFU=.320*(payloadcost_USCM_TFU+...
    Spacecraftcost_USCM_TFU+IAT_USCM_TFU);
LOOS_USCM_TFU=5850; %Launch & Orbital Operations Support (LOOS)%

TotalCost_USCM_TFU=payloadcost_USCM_TFU+Spacecraftcost_USCM_TFU...
    +...
    IAT_USCM_TFU+ProgramLevel_USCM_TFU+LOOS_USCM_TFU;

c(1)=TotalCost_USCM_TFU;


%Calculate SSCM Non-Recurring Engineering (NRE) cost, SME-SMAD 301...
    table 11
%   Percentages from historical values, old SMAD 799
%   "SSCM does not distinguish between non-recurring and recurring...
    cost, as
%   is done in USCM8, and so these CERs predict the aggregate of
```

```
%   non-recurring and recurring costs" SME-SMAD 299
Spacecraftcost_SSCM=1064+35.5*(Spacecraftdrymass(2))^1.261;
payloadcost_SSCM=.4*Spacecraftcost_SSCM;
IAT_SSCM=.139*Spacecraftcost_SSCM;
ProgramLevel_SSCM=.229*Spacecraftcost_SSCM;
GSE_SSCM=.066*Spacecraftcost_SSCM;
LOOS_SSCM=.061*Spacecraftcost_SSCM;

TotalCost_SSCM_NRE=Spacecraftcost_SSCM*.6+payloadcost_SSCM*.6+...
    IAT_SSCM*0+...
     ProgramLevel_SSCM*.5+GSE_SSCM*1+LOOS_SSCM*0;

C(2)=TotalCost_SSCM_NRE;

     %Calculate SSCM Recurring Engineering cost
TotalCost_SSCM_TFU=Spacecraftcost_SSCM*.4+payloadcost_SSCM*.4+...
    IAT_SSCM*1+...
     ProgramLevel_SSCM*.5+GSE_SSCM*0+LOOS_SSCM*1;

c(2)=TotalCost_SSCM_TFU;

Cost=0;
for n=1:types
     Cost = Cost+c(n)*nsats(n)+C(n)*z(n);  % satellite cost
end

% % QuickCost Model
% %******************REFERENCES...
    ******************************************
% Ao=[.18,.406]; %(Initial ref sensor aperature diameter)-table ...
    17-7 pg 510 [multispect midIR/imaging spectoral,thematic mapper...
    ]
% Wo=[250,400]; %Weight in kg of reference payloads (multispec ...
    midIR is 270)
% Po=[225,600]; % Ref power, W (170)
```

```
% dro=[2,75]; % Ref data rate, Mbps (8)
% ...
    %*************************************************************************...

% types=numel(x)/nvars;
% for n=1:types % n refers to sat type
%     nsats(n)=x(n*nvars-nvars+1);
%     diam(n)=x(n*nvars);
%
%     if nsats(n)>0 z(n)=1; else z(n)=0; end
%
%     %system sizing summary
%     Ai(n)=diam(n);  %(New instrument aperture diameter) - sec ...
    17.2.6 pg 516
%
%     R(n)=Ai(n)/Ao(1);  %Sizing Ratio, imaging spectoral p/l
%     if R(n)<.5; K(n)=2; else K(n)=1; end
%     Wi(n)=K(n)*R(n)^3*Wo(1); % design sensor mass (kg)
%     Pi(n)=K(n)*R(n)^3*Po(1); % design sensor power (W)
%     Spacecraftdrymass(n)=(Wi(n)/.31); %Table 14-18 pg 422, LEO w...
    / prop
%
%     M(n)=Wi(n)+Spacecraftdrymass(n); % dry mass instruments + s/...
    c
%     D=0.5; % Percentile relative to state-of-the-art data rate; ...
    average
%     L=12; % Design life in months
%     N=0.3; % Percent new technology; simple Mod
%     Planetary=0; % Earth-orbiting mission
%     Y=2010; % Assume program authority to proceed in 2010
%     Complex=0.5; % Instrument complexity percentile; average
%     T=2; % Team experience; Mixed
%
%     C(n)=2.829*M(n)^0.457*Pi(n)^0.157* 2.718^(0.171*D)* ...
    2.718^(0.00209*L)* 2.718^(1.52*N)*...
```

```
%          2.718^(.258*Planetary)* 1/(2.718^(.0145*(Y-1960)))* ...
    2.718^(.467*Complex)*1/(2.718^(.237*T));
% end
% Cost=sum(C);
report=fopen('data.txt','a');
fprintf(report,'Cost: %d\n\n',Cost);
fclose(report);
```

Listing D.4: Appendix1/constraint.m

```
% Disaggregated Space System Optimization constraint function


function [G, ceq] = constraint(x)
% x = [ 3        1          180     200   1e-6   98.6   0    0    20 ...
      .2;
%       7        1          180     200   1e-6   98.6   0    0    20 ...
      .05]
x
[cov,niir,rev]=gascenario(x)
con=[cov,niir,rev];

ceq = [];
G = [-con+[95,3*ones(1,25),0],...% constrained to >95% of targets,...
    NIIRS>3 (ref45,47), revisit>0min
    con-[101,20*ones(1,25),10]] % constrained to <101% of targets ...
       (impossible), NIIRS<100 (impossible), revisit<60min

IterX=x;
save('IterX.mat','IterX')
```

Listing D.5: Appendix1/Sensor_performance.m

```
% Sensor performance calculation
function NIIRS=Sensor_performance(alt,diam,close_range)
% Replace altitude with closest approach slant range to get NIIRS ...
    for
```

```
% individual targets
global angles

% %% For Example Use
% clear;clc;
% alt=[450,450];
% diam=[0.2,0.5];
% close_range=450;

%% ***************REFERENCES...
    ************************************************
% Declare relevant constants
% IAmax=deg2rad(70);  %IA max is equivalent to max incidence angle
% radiusofearth=6378.1366; %radius of earth (RE) in km from SMAD ...
    pg173
% h=6.6256e-34; %Planck's constant, W*s^2
% c=3e8; % speed of light, m/s
delRho=0.08; % assumed delta reflectance (GIQE users guide, ...
    RefWorks46)
rho=0.15; % assumed reflectance (GIQE users guide, RefWorks46)
K=1.63e-13; %electrons*photons^-1*m^2*sr, assumed VNIR default ...
    value of constant K for SNR, RefWorks46

% Sensor parameters
d=30E-6; %width of focal plane detector in meters; Teledyne ...
    Imaging Sensors CHROMA Visible-Infrared Focal Plane Array
f=10; % meters; assumed focal length (same as Kestrel Eye, ...
    RefWorks47)
Nm=256; %number of focal plane array pixels
lambda=.65E-6; %detector operating wavelenth (lambda) set at mean ...
    wavelength for VNIR

%...
    ************************************************************************...
```

```
% KestrelEye (ref47)
% f=10m
% GSD=1.5m (nadir pointing)
% m=25kg
% alt=450km
%...
    ************************************************************************...


for t=1:numel(close_range)
    SlantRange = close_range(t)*1000;
    for n=1:numel(alt)
        % Determine GSD (validated with KestrelEye stats)
        psi(n)=acos(alt(n)*1000/SlantRange);
        GSD(n)=SlantRange*(d/f)/sqrt(cos(psi(n)));

        % Determine Optical Q // Unused???
        Q=(f*lambda/diam(n))/d; % optical Q, RefWorks46; IFOV=d/f ...
            -> Q=lamb/D/IFOV

        % Determine RER using MTF
        %   *Assume no jitter, smear, or wavefront aberration
        Xsampling=f/d;
        %MTFsampling=sin(pi*X)/(pi*X);
        %   *Assume circular aperture, no obscuration
        Xdiffraction=diam(n)/lambda;
        q=Xsampling/Xdiffraction; % ratio of sampling to diff ...
            cutoff frequency
        %MTFdiffraction=(2/pi)*(acos(X/q)-(X/q)*sqrt(1-(X/q)^2)); ...
            % normalized on sampling MTF

        %MTFsystem=(sin(pi*X)/(pi*X))*((2/pi)*(acos(X/q)-(X/q)*...
            sqrt(1-(X/q)^2))); % MTFsystem=MTFsampling*...
            MTFdiffraction
```

```
%         func=@(X)(((sin(pi*X)./(pi*X)).*((2/pi)*(acos(X/q)-(X/q)...
    .*sqrt(1-(X/q).^2))))).*(sin(pi.*X)/(pi.*X));
%         RER=abs(2*integral(func,0,1/q));
59
        % Determine H as peak value of ER over zeta=1:3, or 1.25 ...
            if
        % monotonical increase
        z=0;
        for zeta=1:.01:3
64          z=z+1;
            func2=@(X)(((sin(pi.*X)./(pi.*X)).*((2/pi).*(acos(X/q)...
                -(X/q).*sqrt(1-(X/q).^2)))./X).*sin(2*pi*X*zeta);
            ER(z)=.5+1/pi*integral(func2,0,1/q);
        end
        if ER(z)-ER(z-50)==ER(z-75)-ER(z-125) % monotonically ...
            increases
69          H(n)=abs(ER(125));
        else
            H(n)=abs(max(ER));
        end

74      zeta=0.5;func2=@(X)(((sin(pi.*X)./(pi.*X)).*((2/pi).*(acos...
            (X/q)-(X/q).*sqrt(1-(X/q).^2)))./X).*sin(2*pi*X*zeta);
        ERplus=.5+1/pi*integral(func2,0,1/q);
        zeta=-0.5;func2=@(X)(((sin(pi.*X)./(pi.*X)).*((2/pi).*(...
            acos(X/q)-(X/q).*sqrt(1-(X/q).^2)))./X).*sin(2*pi*X*...
            zeta);
        ERminus=.5+1/pi*integral(func2,0,1/q);
        RER(n)=abs(ERplus-ERminus);
79
        % Determine SNR
        Ti=0.001; % integration time; conservative estimate from ...
            ref46
%         R_s=integral(@(lamb)lamb/(h*c)*(Ldir+Ldiff),.4,.9); %Vis...
    range
```

```
%         R_h=integral(@(lamb)lamb/(h*c)*(Lpath),.4,.9); %Vis ...
    range
84          % Assuming Sun and sensor are both directly overhead, ...
               approximating
            % data from ref46:
            R_s=0.5*(0.9-0.4)*(1.35e21-6e20)+(0.9-0.4)*6e20;
            R_h=0.5*(0.9-0.4)*2e20;
            SNR(n)=delRho*R_s/sqrt(rho*R_s+R_h)*sqrt(K*Ti);
89
            % Determine G
            % Determine G
            G(n)=10.66; % mean value, rochester.edu source (ref56)
            % Inaccurate conservative estimate is SNR/2, ref46
94


            % GIQE 4 coefficients, metric units (RER<.9), ref46
            c0=5.205;c1=-3.16;c2=2.817;c3=-.656;c4=-.344;


99
            NIIRS(n) = c0+c1*log(GSD(n))+c2*log(RER(n))+c3*H(n)+c4*G(n...
               )/SNR(n); % NIIRS for target t by sensor n


            IFOV=d/f; % instantaneous field of view in radians, ref46
            FOV(n)=IFOV*Nm;
104     end
        NIIRS(t)=max(NIIRS); % best NIIRS achievable by any satellite ...
           during scenario
        fov(t,:)=rad2deg([FOV(1),FOV(2)]);
    end
    % [alt diam close_range]
109 % NIIRS

    % Design angles [ conic ha                        elevation ]
    angles=         [  max(fov(:,1))/2                  90;  % big
                       max(fov(:,2))/2                  90]; % small
```

```
114             %%% ELEVATION IS DEFINED AS: %%%%%%%%%%%%%%%%%%%
                %          90 DEG POINTS NADIR                    %
                %           0 DEG POINTS OUT TO SPACE (SIDEWAYS) %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

119 end
```

Listing D.6: Appendix1/gascenario.m

```
  1 function [Coverage,NIIRS,max_revisit] = gascenario(x)
    % User should check scenario settings, and enter Walker ...
       constellation
    % details. User must also enter number of satellites per ...
       constellation
    % below the Walker section, so that they can be added to the ...
       coverage
    % computation. Currently this assumes same number of small sats as...
        large
  6 % sats.
    % Date: 27 Nov 2013

    % Input x vector, output percent coverage

 11 % Note: clear all causes the MEX link to close
    global nvars ntargets
    error=0;
    %% Scenario Settings
    % User input ///Uncomment this section for independent use
 16 % x = #planes #sats/plane truanom  RAANinc / alt   e   i     om ...
          w  M   / diam
    %********************change inputs to variables...
       *************************
    types=numel(x)/nvars;
    % 6 Satellite IC's:                                    [ a e i ...
       omega w M ]
    for n=1:types
```

```
21      COE(n*6-5:n*6) = x(n*nvars-nvars+5:n*nvars-nvars+10)+[6378.137...
            0 0 0 0 0];
        % Walker parameters
        nplanes(n)=x(n*nvars-nvars+1); % planes
        nspp(n)=x(n*nvars-nvars+2); % sats/plane
        nsats(n)=nplanes(n)*nspp(n); % sats
26      truan(n)=x(n*nvars-nvars+3); % true anomaly spread
        RAANinc(n)=x(n*nvars-nvars+4); % walker raaninc
        % Sensor diameter
        diam(n) = x(n*nvars);
    end
31
    %...
        **************************************************************************...


    % %% Compute sensor angles (included for information only - angles...
        are
    %                           assumed in this program)
36  % IAmax=[1.221,.0872];  %IA max is equivalent to max incidence ...
        angle
    % radiusofearth=6378140; %radius of earth (RE) in meters (14 SMAD ...
        pg977)
    % Nm=[256 256]; %number of focal plane array pixels
    % ymax=68;
    % for n=1:types
41  %     angularradiusofearth(n)=asin(radiusofearth/(radiusofearth+x(...
        n*nvars-nvars+5)*1000));
    %                   % ang rad of earth (rho) (14 SMAD eq 5-24 pg ...
        113)
    %     spacecraftelevationangle(n)=((pi/2)-IAmax(n)); %min ...
        elevation angle in rads (Epsilon)
    %     nadirangle(n)=asin(cos(spacecraftelevationangle(n))*sin(...
        angularradiusofearth(n)));
```

```
%     earthcentralangle(n)=(pi/2)-nadirangle(n)-...
    spacecraftelevationangle(n);
%     Rs(n)=radiusofearth*(sin(earthcentralangle(n))/sin(...
    nadirangle(n))); %Slant Range (Rs)in kilometers
%     IFOV(n)=ymax/Rs(n); %instantaneous field of view in degrees
%     FOV(n)=IFOV(n)*Nm(n);
% end
% % Design angles [ con h-a           elevation ]
% % ///These should simply be calculated in each program that ...
    needs them///
% angles = [ ((FOV(1))/2)              90; % big
%            ((FOV(2))/2)              90] % small
%               %%% ELEVATION IS DEFINED AS: %%%%%%%%%%%%%%%%%%
%               %          90 DEG POINTS NADIR                  %
%               %           0 DEG POINTS OUT TO SPACE (SIDEWAYS) %
%               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%...
    ************************************************************************...

%% Initialize STK and generate scenario components
path='I:\My Documents\THESIS\STK_Matlab\Imager\Matlab\180minute\...
    TargetDeck.sc';
[uiapp, root]       = STK.OpenScen(path);
%Hide STK and do the analysis
uiapp.Visible = 0;
for n=1:ntargets
    target(n)=root.CurrentScenario.children.Item(strcat('target',...
        num2str(n)));
end

% Satellite constellation
constellation=root.CurrentScen.Children.New('eConstellation','...
    Imager');
% Create sats
```

```
for n=1:types
    if nsats(n)==0;
    else
        sat(n)            = STK.create_sat(strcat(num2str(n),'...
            thType'),root,COE(n*6-5:n*6));
        % Create conic imaging sensors
        fixed(n)           = STK.create_sensor(strcat(num2str(n),'...
            thSensor'), sat(n), 1, [10,90]); % angles are conic h/a...
            , el
        eye(n)            = STK.create_sensor(strcat(num2str(n),'...
            thSensorEye'), sat(n), 1, [.2807,90]); % 1 specifies ...
            conic
        slew=eye(n).AccessConstraints.AddNamedConstraint('...
            AngularRate');
        slew.EnableMax=1;slew.Max=1; % Angular rate constraint 1 ...
            deg/sec max slew
        % Add reconnaissance characteristics -> http://www.agi.com...
            /downloads/training/manuals/Fundamentals.pdf
        eye(n).CommonTasks.SetPointingTargetedTracking(2,1,strcat(...
            '*/Target/target1'));
        for m=2:ntargets
            eye(n).Pointing.Targets.Add(strcat('*/Target/target',...
                num2str(m)));
        end
        eye(n).Pointing.EnableAccessTimes=0;
        try
            eye(n).Pointing.ScheduleTimes.Deconflict;
        catch
            error=91;
        end
    end
end

% Create Walker constellation from seed satellite
```

```
% Command inputs: Walker <SatObjectPath> [{WalkerType}] <NumPlanes...
    > <NumSatsPerPlane> {<InterPlaneSpacing> | <TrueAnomaly>} <...
    RAANSpread> {ColorByPlaneFlag} [{AdditionalOptions}]
% Use Walker type ModDelta to enter True Anomaly instead of ...
    Interplane Spacing
for n=1:types
    if nsats(n)==0
    else
        if nsats(n)>1 % only use Walker function for 2 or more ...
            sats
            string=horzcat('Walker */Satellite/',num2str(n),'...
                thType ModDelta ',num2str(nplanes(n)),' ',num2str(...
                nspp(n)),' ',num2str(truan(n)),' ',num2str(RAANinc(...
                n)),' Yes');
            root.ExecuteCommand(string);
            % Keep things simple - stay UNDER 10 planes, 10 sats (...
                need to
            % adjust for 10 spp)
            for p=1:nplanes(n)
                for m=1:nspp(n)
                    path1=horzcat('Satellite/',num2str(n),'thType'...
                        ,num2str(p),num2str(m),'/Sensor/',num2str(n...
                        ),'thSensorEye');
                    path2=horzcat('Satellite/',num2str(n),'thType'...
                        ,num2str(p),num2str(m),'/Sensor/',num2str(n...
                        ),'thSensor');
                    try
                        constellation.Objects.Add(path1);
                        constellation.Objects.Add(path2);
                    catch
                        error=111;
                    end
                end
            end
        else
```

```
            string1=horzcat('Satellite/',num2str(n),'thType/Sensor...
                /',num2str(n),'thSensor');
            string2=horzcat('Satellite/',num2str(n),'thType/Sensor...
                /',num2str(n),'thSensorEye');
            constellation.Objects.Add(string1);
            constellation.Objects.Add(string2);
            for n=1:ntargets
                Cov=horzcat('Cov */Target/target',n,' Asset */',...
                    string2,' Assign');
                try
                    root.ExecuteCommand(Cov);
                catch
                    error=129;
                end
            end
        end
    end

    %% Determine figures of merit
go=0;
    if sum(nsats)>0
        m=1;
        for n=1:ntargets
            % Slant range (and access boolean)
            cov=target(n).ObjectCoverage;
            cov.Assets.Add('Constellation/Imager');
            cov.FOM.SetAccessConstraintDefinition(20); % access ...
                constraint, range
            cov.FOM.Definition.AcrossAssets='eFmMinimum';
            cov.FOM.Definition.SetComputeType('eMinimum');
            cov.Compute;
            stuff=root.ExecuteCommand(strcat('Cov_RM */Target/target',...
                num2str(n),' Access Compute "FOM Value"'));
            range_file=fopen('range.txt','w+');
```

```
for k=0:stuff.Count-1;fprintf(range_file,'%s\n',stuff.Item...
    (k));end
range_data=importdata('range.txt',',',8);
if numel(range_data)<3
    list=find(range_data.data~=0);
    if numel(list)>0
        close_range(n)=min(range_data.data(list));
        index_close(n)=find(range_data.data==close_range(n...
            ));
        close_range_time(n)=range_data.textdata(8+...
            index_close(n));
        far_range(n)=max(range_data.data);
        index_far(n)=find(range_data.data==far_range(n),1)...
            ;
        far_range_time(n)=range_data.textdata(8+index_far(...
            n));
        fclose('all');
        no_access=[];
        go=1;
    else
        close_range(n)=inf;
        far_range(n)=inf;
    end
else
    disp(strcat('No access for target ',num2str(n)));
    no_access(m)=n;
    close_range(n)=inf;
    far_range(n)=inf;
    m=m+1;
end
% Revisit time (should have longer scenario time than ~8 ...
    hrs)
cov.FOM.SetDefinitionType(10); %revisit time
cov.Compute;
```

```
        stuff=root.ExecuteCommand(strcat('Cov_RM */Target/target',...
            num2str(n),' Access Compute "FOM Value"'));
        revisit_file=fopen('revisit.txt','w+');
        for k=0:stuff.Count-1;fprintf(revisit_file,'%s\n',stuff....
            Item(k));end
        try
181         revisit(n)=importdata('revisit.txt',',',8);
            gap(n)=max(revisit(n).data);
        catch
            gap(n)=0;
        end
186     fclose('all');
    end


    %% Percent coverage for targets
191 if go==1
        Coverage=100*(ntargets-numel(no_access))/ntargets;
        max_revisit=max(gap)/60;

        time=clock;
196     report=fopen('data.txt','a');
        fprintf(report,'Time: %d-%d-%d %d:%d:%d\n',time);
        fprintf(report,'Design vector: %2.0f %2.0f %3.3f %3.3f ...
            %4.3f %1.3f %3.3f %3.3f %3.3f %3.3f %1.3f]\n',x);
        fprintf(report,' Max Revisit Time (min) %3.3f\n',...
            max_revisit);
        fprintf(report,' Max Range %3.3f\n',max(far_range));
201     fprintf(report,' Min Range %3.3f\n',min(close_range(find(...
            close_range~=0))));
        fprintf(report,' Target Coverage (percent) %3.3f\n',...
            Coverage);
        fprintf(report,'Error=%1.0f\n\n',error);
        fclose('all');
    else
```

```
206             Coverage=0;
                max_revisit=inf;
            end
        else
            close_range=inf(1,ntargets);
211         Coverage=0;
            max_revisit=inf;
        end

        %% Sensor performance calculation
216     % Replace altitude with closest approach slant range to get NIIRS ...
           for
        % individual targets
        for n=1:types
            alt(n) = x(n*nvars-nvars+4)*1000; % alt in meters
            diam(n) = x(n*nvars);
221     end
        NIIRS=Sensor_performance(alt,diam,close_range);
        end
```

D.2 STK Library

Listing D.7: Appendix1/STK.m

```
    classdef STK
  2 %%STK Library created by James Sales
    %
    % This library serves as a conduit between STK and MATLAB.  It is ...
       generally
    % geared towards my specific research but has been designed to be ...
       useful
    % elsewhere for the most part.  Below is a list of the structure ...
       fields
  7 % used in the various functions defined in this library.
    %
    % Scen Structure Fields:
```

```
%       Centroid:       The Lattitude, Longitude, and Elevation of...
 the
%                       desired ellipse for an Area Target.
%       COE:            The Initial State Classical Orbital ...
Elements
%                       formatted as follows:
%                           [r_p    e    i    RAAN    w    nu]
%                       the Radius of Periapsis is in kilometers ...
and all
%                       angles are in degrees.
%       ElevAngle:      Minimum Elevation Angle for Access to ...
satellite.
%       EndTime:        The Scen end time formatted as follows:
%                           'DD MMM YYYY HH:MM:SS'
%       EngineName:     String for the desired engine name.
%       Epoch:          The Epoch time formatted as follows:
%                           'DD MMM YYYY HH:MM:SS'
%       m_sat:          The satellite dry mass in kg.
%       m_fuel:         The fuel mass in kg.
%       Now:            Tracks time from Epoch to current maneuver...
 in
%                       seconds.
%       Path:           The filepath for external file storage.
%       Prop:           The desired propagation engine.
%       Size:           The semi-major axis, semi-minor axis, and ...
bearing
%                       formatted as a vector for the desired ...
ellipse for
%                       an Area Target.
%       StartTime:      The Scen start time formatted as follows:
%                           'DD MMM YYYY HH:MM:SS'
%       TimeStep:       Animation increment given in seconds.
%       Title:          A string describing the desired Scen title...
.
%                       This string must contain no spaces.
```

```
37 methods(Static)
       function [uiapp, root] = Initialize(Scen)
           % This function initializes STK and passes back the ...
              applicable handles for
           % further use in MATLAB.  The function takes the following...
               inputs:
           %
42         %       [uiapp, root] = STK.Initialize(Scen)

           %% Grab STK handle if already running or open STK if not
           try
               uiapp = actxGetRunningServer('STK10.application');
47         catch
               uiapp = actxserver('STK10.application');
           end
           root       = uiapp.Personality2;
           %% Close existing Scen and open a new one
52         try
               root.CloseScenario;
               root.NewScenario(Scen.Title);
           catch
               root.NewScenario(Scen.Title);
57         end
           %% Set Scen Preferences
           %  Set Date/Time Format
           root.UnitPreferences.Item('DateFormat').SetCurrentUnit('...
              UTCG');
           %  Assign Scen time period
62         scen                                = root.CurrentScen;
           scen.SetTimePeriod(Scen.StartTime,Scen.EndTime);
           scen.Animation.StartTime            = Scen.StartTime;
           scen.Epoch                          = Scen.StartTime;
           scen.Animation.AnimStepValue        = Scen.TimeStep;
67         %% Set Animation to Start Time
           root.Rewind()
```

```
end

function [uiapp, root] = OpenScen(path)
    % This function initializes STK and passes back the ...
       applicable handles for
    % further use in MATLAB of an existing scenario.
    % The function takes the following inputs:
    %
    %       [uiapp, root] = STK.Initialize(path)

    %% Grab STK handle if already running or open STK if not
    try
        uiapp = actxGetRunningServer('STK10.application');
    catch
        uiapp = actxserver('STK10.application');
    end
    root      = uiapp.Personality2;
    %% Close existing Scen and open a new one
    try
        root.CloseScenario;
        root.LoadScenario(path);
    catch
        root.LoadScenario(path);
    end
    %% Set Animation to Start Time
    root.Rewind()
end

function [sat] = create_sat(Name, root, COE)
    % This function initializes a satellite in Astrogator and ...
       returns the
    % applicable handles for further use in MATLAB.  It takes ...
       the following
    % inputs:
    %
```

```
        %       [sat] = create_sat(Name, root, COE)
102
        %% Initialize Satellite
        missionStartDate              = root.CurrentScen.StartTime;
        sat                           = root.CurrentScen.Children....
            New(18, Name);

107     %% Define the Initial States
        % Create handle to the Initial States
        IS = sat.Propagator.InitialState;

        % Input orbital elements
112     IS.Representation.AssignClassical('eCoordinateSystemICRF'...
            ,...
             COE(1),COE(2),COE(3),COE(4),COE(5),COE(6));
        % ^ See STK programming interface help, ...
            AgECoordinateSystem Enumeration
        % ICRF seems to be the default

117     % Sets the orbit Epoch for the mission start time
        sat.Propagator.StartTime                    = ...
            missionStartDate;
        % Propagate satellite
        sat.Propagator.Propagate
    end
122
    function [target] = create_target(Name, root, location)
        % This function creates a target in the scenario at the ...
            coordinates
        % specified, of format:
        %           [lat,long,alt]
127
        target=root.CurrentScen.Children.New('eTarget',Name);
        target.position.AssignGeodetic(location(1),location(2),...
            location(3))
```

```
    end

132 function [sat, MCS] = Astrogator(Name, root, Scen)
    % This function initializes a satellite in Astrogator and ...
       returns the
    % applicable handles for further use in MATLAB.  It takes ...
       the following
    % inputs:
    %
137 %       [sat, MCS] = STK.Astrogator(Name, root, Scen)

    %% Initialize Satellite
    missionStartDate             = root.CurrentScen.StartTime;
    sat                          = root.CurrentScen.Children....
       New(18, Name);
142 sat.SetPropagatorType('ePropagatorAstrogator')
    sat.Graphics.Attributes.Intervals.RemoveAll;
    sat.Graphics.Attributes.Default.Inherit = 0;
    sat.Graphics.Attributes.Default.IsOrbitVisible = 0;
    % Create handle to the Astrogator portion of the satellite...
       's object model
147 prop                         = sat.Propagator;
    % Create handle to the MCS and remove all existing ...
       segments
    MCS                          = prop.MainSequence;
    MCS.RemoveAll;
    %% Define the Initial States
152 % Create handle to the Initial States
    IS = MCS.Insert('eVASegmentTypeInitialState','Initial ...
       State','-');
    % Designate satellite and fuel masses
    IS.SpacecraftParameters.DryMass = Scen.m_sat;
    IS.FuelTank.FuelMass            = Scen.m_fuel;
157 IS.FuelTank.MaximumFuelMass     = Scen.m_fuel;
    % Input orbital elements
```

```
        IS.SetElementType('eVAElementTypeModKeplerian');
        IS.Element.RadiusOfPeriapsis    = Scen.COE(1);
        IS.Element.Eccentricity         = Scen.COE(2);
        IS.Element.Inclination          = Scen.COE(3);
        IS.Element.RAAN                 = Scen.COE(4);
        IS.Element.ArgOfPeriapsis       = Scen.COE(5);
        IS.Element.TrueAnomaly          = Scen.COE(6);
        % Sets the orbit Epoch for the mission start time
        IS.OrbitEpoch                   = missionStartDate;
    end

    function [sat] = Create_Satellite_External(Name, root, path)
        % This function initializes a satellite with an external ...
            ephemeris
        % file and returns the applicable handle for further use ...
            in MATLAB.
        % It takes the following inputs:
        %
        %       sat = STK.Create_Satellite_External(Name, root, ...
            path)

        %% Initialize Satellite
        sat                         = root.CurrentScen.Children....
            New(18, Name);
        sat.SetPropagatorType('ePropagatorStkExternal');
        %% Specify filepath for ephemeris file
        prop                        = sat.Propagator;
        prop.filename               = path;
        %% Propagate satellite
        prop.Propagate;
    end

    function [Target] = Area_Target(Name, root, Scen)
        % This function initializes an Area Target in STK and ...
            returns the
```

```
        % applicable handles for further use in MATLAB.  It takes ...
            the following
        % inputs:
        %
192     %       Target = STK.Area_Target(Name, root, Scen)

        Size     = Scen.Size;
        Centroid = Scen.Centroid;
        Target   = root.CurrentScen.Children.New(2, Name);
197     Target.AreaType = 'eEllipse';
        Target.AreaTypeData.SemiMajorAxis = Size(1);
        Target.AreaTypeData.SemiMinorAxis = Size(2);
        Target.AreaTypeData.Bearing       = Size(3);
        Target.Position.AssignGeodetic(Centroid(1),Centroid(2),...
            Centroid(3));
202     Target.AccessConstraints.AddNamedConstraint('...
            ElevationAngle');
        Target.AccessConstraints.GetActiveNamedConstraint('...
            ElevationAngle').Angle = Scen.ElevAngle;
    end

    function [coverage] = coverage_definition(Name, root)
207     % This function initializes a coverage definition in STK ...
            and returns the
        % applicable handles for further use in MATLAB.  It takes ...
            the following
        % inputs:
        %
        %       coverage = STK.coverage_definition(Name, root)
212

        coverage   = root.CurrentScen.Children.New('...
            eCoverageDefinition', Name);
        coverage.Grid.BoundsType = 'eBoundsGlobal';
    end
```

```
217
        function [FOM] = create_FOM(Name, coverage)
            % This function initializes a FOM definition in STK and ...
               returns the
            % applicable handles for further use in MATLAB.  It takes ...
               the following
            % inputs:
222         %
            %       FOM = STK.create_FOM(Name, coverage)


            FOM    = coverage.Children.New('eFigureOfMerit', Name);
227     end

        function [sensor] = create_sensor(Name, sat, type, angle)
            % This function initializes a RECTANGULAR sensor object ...
               in STK and returns the
            % applicable handles for further use in MATLAB.  It takes...
                the following
232         % inputs:
            %
            %       sensor = STK.create_sensor(Name, sat, type, angle...
               )
            %
            %        Type:
237         %      1 = simple conic
            %          angle should be 1x2 vector, in deg
            %          [ conic half angle, nadir angle ]
            %
            %      2 = rectangular
242         %          angle should be 1x3 vector in degrees ...
               formatted:
            %          vertical halfangle, horizontal halfangle, ...
               nadir angle
```

```
        % This section can be updated to include other types of ...
            sensor
        % characteristics; this is specifically for the GA ...
            scenario
        % -2Lt Evelyn Abbate, 4 Oct 2013


        sensor   = sat.Children.New('eSensor', Name);
        switch type
            case 1
                sensor.Pattern.ConeAngle=angle(1);
                sensor.FocalLength=10;
                sensor.CommonTasks.SetPointingFixedAzEl(0,angle...
                    (2),1);
            case 2
                sensor.SetPatternType(3);
                sensor.Pattern.VerticalHalfAngle=angle(1);
                sensor.Pattern.HorizontalHalfAngle=angle(2);
                sensor.CommonTasks.SetPointingFixedAzEl(0,angle...
                    (3),1);
        end
    end


    function [Eng] = Create_Engine_Model(root, Name, T)
        % This function creates a custom engine model in the ...
            Componant Library and
        % returns the applicable handle for further use in MATLAB....
            It takes the
        % following inputs:
        %
        %       Eng = STK.CreateEngingModel(root, Name, T)

        scen         = root.CurrentScen;
```

```
        EM           = scen.ComponentDirectory.GetComponents('...
            eComponentAstrogator').GetFolder('Engine Models');
        ConstThrust  = EM.Item('Constant Thrust and Isp');
        ConstThrust.CloneObject;
        num          = EM.count;
277     for count = 0:num-1
            if length(EM.Item(count).Name) > 23
                if strcmp(EM.Item(count).Name(1:24),'Constant ...
                    Thrust and Isp1')
                    Eng = EM.Item(count);
                end
282         end
        end
        Eng.Name     = Name;
        Eng.Thrust   = T;
    end
287
    function [prop] = Propagate(Name, t, MCS, Scen)
        % This function adds a propagation step to the given ...
            satellite in
        % Astrogator and returns the applicable handle for further...
            use in MATLAB.
        % It takes inputs as follows:
292     %
        %       prop = STK.Propagate(Name, t, MCS, Scen)

        prop = MCS.Insert('eVASegmentTypePropagate',Name,'-');
        prop.PropagatorName = Scen.Prop;
297     prop.StoppingConditions.Item('Duration').Properties.Trip =...
            t;
    end

    function [AccessTimes] = Compute_Access(root, sat, target, ...
        clock)
```

```
        % This function takes two handles and computes coverage ...
            encounters over
302     % the entire Scen.  It takes the following inputs:
        %
        %       AccessTimes = STK.Compute_Access(root, sat, target...
            , count)

        root.UnitPreferences.Item('DateFormat').SetCurrentUnit('...
            EpSec');
307     scen       = root.CurrentScen;
        access     = target.GetAccessToObject(sat);
        access.ComputeAccess;
        DP         = access.DataProviders.Item('Access Data').Exec...
            (scen.StartTime, scen.StopTime);
        Enter      = cell2mat(DP.DataSets.GetDataSetByName('Start ...
            Time').GetValues);
312     Depart     = cell2mat(DP.DataSets.GetDataSetByName('Stop ...
            Time').GetValues);
        for count = 1:min(length(Enter),length(Depart))
            Entry(count,:) = RO.Time_Sequencer(clock, Enter(count)...
                );
            Exit(count,:) = RO.Time_Sequencer(clock, Depart(count)...
                );
            Spaces(count,:) = '     ';
317     end
        AccessTimes.DT    = [Entry Spaces Exit];
        AccessTimes.EpSec = [Enter Depart];
    end

322 function [t_end] = Output_to_text(Scen, Out, L, CoastTime)
        % This function generates a text file conforming to the ...
            Astrogator *.a
        % thrust attitude external file input parameters.  It ...
            takes inputs as
        % follows:
```

```
%
%       t_end = STK.Out_to_text(Scen, Out, L, CoastTime)
%
% Out Structure Fields:
%       length:         length of the time vector
%       t:              The time vector in seconds
%       ECI:            The Earth-Centered Inertial ...
    attitude vector

Filename = [Scen.Path,Scen.Title,'Profile',num2str(count),...
    '.a'];
t        = Out.t+Scen.Now;
t_end    = Out.t(end)-CoastTime+Scen.Now;
ECI      = Out.ECI;
Epoch    = Scen.Epoch;
Maneuver = [t ECI]';
Points   = length(t)-L;
Factor   = 20;
Order    = 1;
Body     = 'Earth';
Axes     = 'Inertial';
% Open file & begin writing data conforming to the STK ...
    format requirements.
FID = fopen(Filename,'w');
fprintf(FID,'stk.v.5.0\r\n \r\n');
fprintf(FID,'BEGIN Attitude\r\n \r\n');
fprintf(FID,'NumberOfAttitudePoints\t%1.0f\r\n',Points);
fprintf(FID,['Scen Epoch\t\t',Epoch,'\r\n']);
fprintf(FID,'Blocking Factor\t\t%2.0f\r\n',Factor);
fprintf(FID,'InterpolationOrder\t%1.0f\r\n',Order);
fprintf(FID,['CentralBody\t\t',Body,'\r\n']);
fprintf(FID,['CoordinateAxes\t\t',Axes,'\r\n\r\n']);
fprintf(FID,'AttitudeTimeECIVector\r\n\r\n');
fprintf(FID,'\t%6.6f \t\t%8.8f \t\t%8.8f \t\t%8.8f \r\n',...
    Maneuver(:,L+1:end));
```

```
357         fprintf(FID,'\r\nEND Attitude');
            fclose('all');
        end

        function [] = Maneuver_From_File(Name, MCS, t, Scen, index)
362         % This function conducts a Finite Thrust Vectored manuever...
                in Astrogator
            % and returns the applicable maneuver handle for further ...
               use in MATLAB.
            % It takes inputs as follows:
            %
            %       STK.Maneuver_From_File(Name, MCS, t, Scen, index)
367
            Filename = [Scen.Path,Scen.Title,'Profile',num2str(index),...
               '.a'];
            M = MCS.Insert('eVASegmentTypeManeuver',Name,'-');
            M.SetManeuverType('eVAManeuverTypeFinite');
            M.Maneuver.SetAttitudeControlType('eVAAttitudeControlFile'...
               );
372         Att_Control = M.Maneuver.AttitudeControl;
            Att_Control.Filename = Filename;
            M.Maneuver.SetPropulsionMethod('...
               eVAPropulsionMethodEngineModel', Scen.EngineName);
            M.Maneuver.Propagator.StoppingConditions.Item('Duration')....
               Properties.Trip = t;
            M.Maneuver.Propagator.PropagatorName = Scen.Prop;
377     end

        function [] = FTV_Maneuver(Name, MCS, v, t)
            % This function conducts a Finite Thrust Vectored manuever...
                in Astrogator
            % and returns the applicable maneuver handle for further ...
               use in MATLAB.
382         % It takes inputs as follows:
            %
```

```
        %       STK.FTV_Maneuver(Name, MCS_root, Vector, Duration)

        global Scen
387     M = MCS.Insert('eVASegmentTypeManeuver',Name,'-');
        M.SetManeuverType('eVAManeuverTypeFinite');
        M.Maneuver.SetAttitudeControlType('...
            eVAAttitudeControlThrustVector');
        Att_Control = M.Maneuver.AttitudeControl;
        Att_Control.ThrustVector.AssignXYZ(v(1),v(2),v(3));
392     M.Maneuver.SetPropulsionMethod('...
            eVAPropulsionMethodEngineModel', Scen.EngineName);
        M.Maneuver.Propagator.StoppingConditions.Item('Duration')....
            Properties.Trip=t;
        M.Maneuver.Propagator.PropagatorName = Scen.Prop;
    end

397 function [] = ITV_Maneuver(Name, MCS, v)
        % This function conducts an Impulsive Thrust Vectored ...
            manuever in
        % Astrogator and returns the applicable maneuver handle ...
            for further use in
        % MATLAB.  It takes inputs as follows:
        %
402     %       STK.ITV_Maneuver(Name, MCS, Vector)

        M = MCS.Insert('eVASegmentTypeManeuver',Name,'-');
        M.Maneuver.SetAttitudeControlType('...
            eVAAttitudeControlThrustVector');
        Att_Control = M.Maneuver.AttitudeControl;
407     Att_Control.DeltaVVector.AssignCartesian(v(1),v(2),v(3));
    end

    function [t, Elem] = Elements(sat, time, Type)
        % This function takes a satellite and returns its orbital ...
            element time
```

```
412         % history.  It takes the following inputs:
            %
            %       [t, Elem] = STK.Elements(sat, time, Type)

            root = sat.root;
417         root.UnitPreferences.SetCurrentUnit('DateFormat','EpSec');
            if Type == 'E'
                EE          = sat.DataProviders.Item('Equinoctial ...
                   Elements');
                EEICRF      = EE.Group.Item('ICRF');
                EEResults   = EEICRF.Exec(time(1), time(2), 20);
422             t = cell2mat(EEResults.DataSets.GetDataSetByName('Time...
                   ').GetValues());
                a = cell2mat(EEResults.DataSets.GetDataSetByName('Semi...
                   -Major Axis').GetValues());
                h = cell2mat(EEResults.DataSets.GetDataSetByName('e * ...
                   sin(omegaBar)').GetValues());
                k = cell2mat(EEResults.DataSets.GetDataSetByName('e * ...
                   cos(omegaBar)').GetValues());
                p = cell2mat(EEResults.DataSets.GetDataSetByName('tan(...
                   i/2) * sin(raan)').GetValues());
427             q = cell2mat(EEResults.DataSets.GetDataSetByName('tan(...
                   i/2) * cos(raan)').GetValues());
                F = cell2mat(EEResults.DataSets.GetDataSetByName('Mean...
                    Lon').GetValues());
                Elem = [a h k p q F];
            elseif Type == 'C'
                COE         = sat.DataProviders.Item('Classical ...
                   Elements');
432             COEICRF     = COE.Group.Item('ICRF');
                COEResults = COEICRF.Exec(time(1),time(2),20);
                t = cell2mat(COEResults.DataSets.GetDataSetByName('...
                   Time').GetValues());
                a = cell2mat(COEResults.DataSets.GetDataSetByName('...
                   Semi-major Axis').GetValues());
```

```
            e = cell2mat(COEResults.DataSets.GetDataSetByName('...
                Eccentricity').GetValues());
437         i = cell2mat(COEResults.DataSets.GetDataSetByName('...
                Inclination').GetValues());
            omega = cell2mat(COEResults.DataSets.GetDataSetByName(...
                'RAAN').GetValues());
            w = cell2mat(COEResults.DataSets.GetDataSetByName('Arg...
                 of Perigee').GetValues());
            M = cell2mat(COEResults.DataSets.GetDataSetByName('...
                Mean Anomaly').GetValues());
            lat = cell2mat(COEResults.DataSets.GetDataSetByName('...
                Arg of Latitude').GetValues());
442         nu = cell2mat(COEResults.DataSets.GetDataSetByName('...
                True Anomaly').GetValues());
            Elem = [a e i omega w M lat nu];
        else
            t = [];
            Elem = [];
447         fprintf('Specified Type not recognized\n')
        end
    end

    function[] = Export_Ephemeris(sat, Start, Stop, path)
452     % This function generates an ephemeris export file for the...
            specified
        % satellite.  It takes the following inputs:
        %
        %       STK.Export_Ephemeris(sat, Start, Stop, path)
        %
457     % where sat is a handle specifying the satellite for which...
            the
        % ephemeris file is desired, Start and Stop designate the ...
           start and
        % stop times for the ephemeris file, and path designates ...
           the filepath
```

```
            % for where this file is exported.
            %
462         % Start and Stop should be a string formatted as follows:
            %
            %       dd mmm yyyy hh:mm:ss
            %
            % Path should be formatted as a string and at a minimum ...
               specify
467         % the desired filename ending in '.e'.  If no path is ...
               specified,
            % the file will be exported to the default STK folder.

            %% Get ephemeris handle
            stkEphem = sat.ExportTools.GetEphemerisStkExportTool;
472         %% Specify time period
            stkEphem.TimePeriod.TimePeriodType = '...
               eExportToolTimePeriodSpecify';
            stkEphem.TimePeriod.Start = Start;
            stkEphem.TimePeriod.Stop = Stop;
            stkEphem.StepSize.StepSizeType = 'eExportToolStepSizeEphem...
               ';
477         %% Export ephemeris file to designated location
            stkEphem.Export(path);
        end
    end
    end
```

This Appendix contains the plots generated by the GA showing how the algorithm closed in on solutions for the cheapest and fastest cases, and the associated populations.

E.1 Middle East Target Deck

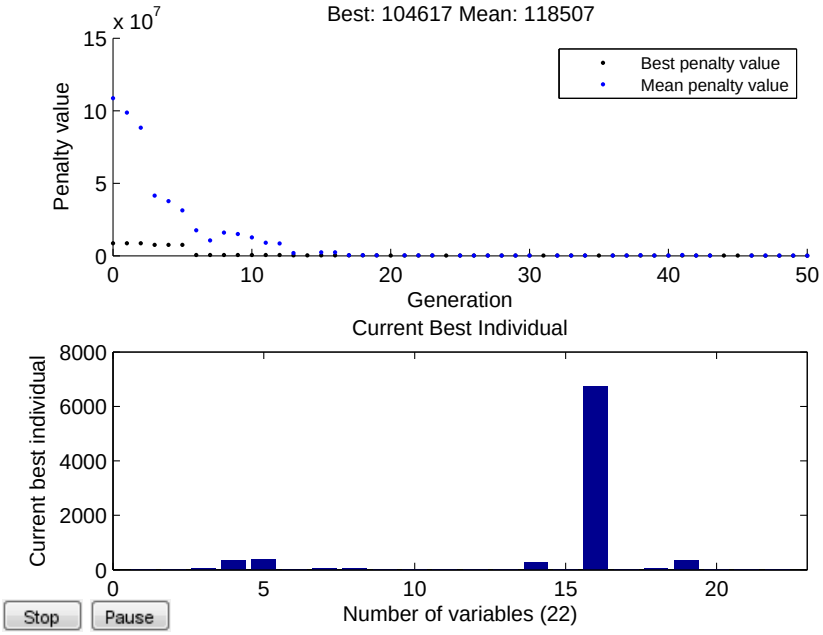


Figure E.1: Middle East Cheapest Solution. GA output plot for Middle East target deck with 30 minute constraint. Actual revisit time is 79 minutes.

MIDDLE EAST - Cheapest

		scores	population										
Revisit time	Generations	cost (millions)	nplanes	spp	raan inc	truanspr	alt	ecc	incl	raan	argp	M	diam (m)
		$ 433.46	1	1	188.0307	324.4855	1462.664	1.00E-06	31.39443	214.5387	1.00E-06	20	0.147455
			42	1	318.1196	196.2546	5898.099	1.00E-06	88.47474	92.0406	1.00E-06	20	0.047463
		$ 433.46	1	1	188.0307	324.4855	1462.664	1.00E-06	31.39443	214.5387	1.00E-06	20	0.147455
			42	1	318.1196	196.2546	5898.099	1.00E-06	88.47474	92.0406	1.00E-06	20	0.047463
		$ 433.46	1	1	188.0307	324.4855	1462.664	1.00E-06	31.39443	214.5387	1.00E-06	20	0.147455
			42	1	318.1196	196.2546	5898.099	1.00E-06	88.47474	92.0406	1.00E-06	20	0.047463
		$ 443.44	1	1	185.9623	277.4391	1303.474	1.00E-06	31.13427	331.4713	1.00E-06	20	0.145454
			26	2	334.6793	151.7021	4470.07	1.00E-06	81.57528	165.9094	1.00E-06	20	0.047139
		$ 450.16	1	1	182.4441	290.4053	1302.313	1.00E-06	30.78517	204.9045	1.00E-06	20	0.145152
			27	2	351.3249	150.6934	6035.289	1.00E-06	88.8212	87.15927	1.00E-06	20	0.047767
		$ 2,156.20	1	1	188.0235	324.9039	1292.208	1.00E-06	31.43984	184.9314	1.00E-06	20	0.14748
			14	3	317.792	199.7466	3946.983	1.00E-06	40.54378	79.21177	1.00E-06	20	0.139876
		$ 2,156.20	1	1	188.181	311.0549	1252.362	1.00E-06	31.22896	213.5316	1.00E-06	20	0.147011
			40	3	318.3152	196.3836	5935.166	1.00E-06	80.43737	91.51854	1.00E-06	20	0.087597
		$ 2,156.20	1	1	188.1085	330.8742	1458.895	1.00E-06	31.12743	203.3844	1.00E-06	20	0.14586
			26	18	333.3862	196.2032	4863.972	1.00E-06	88.66974	91.05742	1.00E-06	20	0.04715
		$ 663.03	1	1	182.3191	291.9539	1541.478	1.00E-06	30.99399	358.6415	1.00E-06	20	0.145556
			49	3	312.9133	196.325	5810.476	1.00E-06	89.05766	89.08425	1.00E-06	20	0.04725
		$ 2,156.20	1	1	181.9148	276.5978	1301.932	1.00E-06	31.12868	296.4702	1.00E-06	20	0.14552
			26	2	336.1207	151.7469	5939.254	1.00E-06	88.97954	83.41904	1.00E-06	20	0.103977
		$ 2,156.20	1	1	180.9775	283.2439	1481.541	1.00E-06	31.87192	279.2018	1.00E-06	20	0.261814
			16	4	309.5006	156.5953	4228.77	1.00E-06	84.35798	97.96224	1.00E-06	20	0.056327
		$ 511.27	1	1	186.33	273.0577	1460.045	1.00E-06	31.14914	304.0529	1.00E-06	20	0.14569
			27	3	334.7248	197.6983	4596.668	1.00E-06	88.66498	75.90476	1.00E-06	20	0.047161
		$ 2,156.20	1	1	191.0545	311.2539	1465.524	1.00E-06	31.09347	214.9249	1.00E-06	20	0.145197
			29	1	318.0768	186.4408	5924.78	1.00E-06	88.48441	194.2986	1.00E-06	20	0.049114
		$ 433.46	1	1	188.0307	324.4855	1462.664	1.00E-06	31.39443	214.5387	1.00E-06	20	0.147455
			42	1	318.1196	196.2546	5898.099	1.00E-06	88.47474	92.0406	1.00E-06	20	0.047463
		$ 433.46	1	1	188.0307	324.4855	1462.664	1.00E-06	31.39443	214.5387	1.00E-06	20	0.147455
			42	1	318.1196	196.2546	5898.099	1.00E-06	88.47474	92.0406	1.00E-06	20	0.047463
		$ 2,156.20	1	1	187.5976	314.7135	1391.198	1.00E-06	30.80649	205.092	1.00E-06	20	0.146453
			11	1	320.0775	150.8718	5901.118	1.00E-06	88.53367	88.22424	1.00E-06	20	0.047453
		$ 627.74	1	1	189.681	292.0385	1464.642	1.00E-06	31.03643	201.8866	1.00E-06	20	0.145592
			43	3	318.0388	196.2265	5869.677	1.00E-06	88.50289	91.17318	1.00E-06	20	0.047658
		$ 2,156.20	0	0	182.4441	290.4053	1302.313	1.00E-06	30.78517	204.9045	1.00E-06	20	0.145152
			27	1	351.3249	150.6934	6035.289	1.00E-06	88.8212	87.15927	1.00E-06	20	0.047767
		Inf	10	1	188.2396	324.5286	1463.317	1.00E-06	31.39274	214.7154	1.00E-06	20	0.147452
			45	0	318.1705	196.4536	5891.176	1.00E-06	88.47659	91.92877	1.00E-06	20	0.047463
		$ 2,156.20	1	0	180.4313	288.4043	1321.191	1.00E-06	31.6653	272.8061	1.00E-06	20	0.146227
104	50		21	1	309.5485	156.7316	6253.3	1.00E-06	43.55997	97.9965	1.00E-06	20	0.050951

Table E.1: Population and Scores for Cheapest Solution, Middle East Target Deck

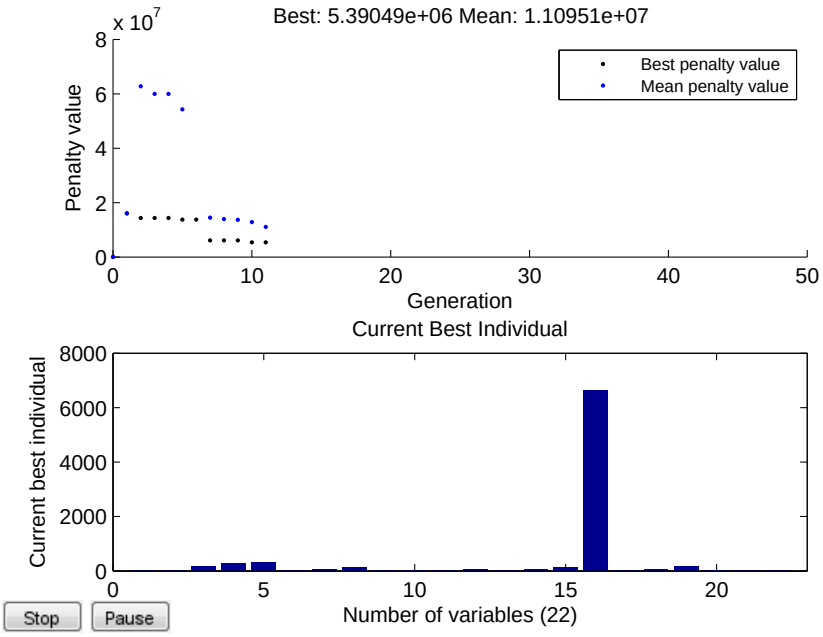


Figure E.2: Middle East Fastest Solution. GA output plot for Middle East target deck with 15 minute constraint. Actual revisit time is 1.89 minutes.

MIDDLE EAST - Fastest													
		scores	population										
Revisit time	Generations	cost (millions)	nplanes	spp	raan inc	truanspr	alt	ecc	incl	raan	argp	M	diam (m)
		$ 5,390.49	1	8	168.8759	269.8837	295.2446	1.00E-06	41.92351	138.8296	1.00E-06	20	0.182408
			40	7	63.63983	131.0282	6628.546	1.00E-06	50.52629	164.0044	1.00E-06	20	0.102582
		$ 6,122.58	1	12	234.1936	265.4981	261.0896	1.00E-06	37.54399	169.5611	1.00E-06	20	0.168898
			43	6	62.58539	130.7609	6588.302	1.00E-06	53.79843	239.4934	1.00E-06	20	0.108151
		$ 6,704.47	1	8	167.4514	270.1474	264.5918	1.00E-06	41.68942	222.3295	1.00E-06	20	0.182257
			38	8	92.86172	226.0807	6415.334	1.00E-06	55.02203	170.9392	1.00E-06	20	0.110339
		$ 7,745.74	1	12	187.2	264.4683	259.9302	1.00E-06	41.65316	221.2035	1.00E-06	20	0.182633
			39	9	64.80634	136.9868	6417.186	1.00E-06	54.90821	157.7206	1.00E-06	20	0.106907
		$ 13,734.36	1	9	164.9576	213.4902	294.7095	1.00E-06	41.69573	262.8768	1.00E-06	20	0.182326
			38	8	97.02846	202.7234	6361.729	1.00E-06	66.89559	173.1741	1.00E-06	20	0.141749
		$ 13,435.46	1	9	164.9576	213.4902	294.7095	1.00E-06	41.69573	262.8768	1.00E-06	20	0.182326
			37	8	97.02846	202.7234	6361.729	1.00E-06	66.89559	173.1741	1.00E-06	20	0.141749
		$ 5,390.49	1	8	168.8759	269.8837	295.2446	1.00E-06	41.92351	138.8296	1.00E-06	20	0.182408
			40	7	63.63983	131.0282	6628.546	1.00E-06	50.52629	164.0044	1.00E-06	20	0.102582
		$ 14,332.16	1	10	165.1497	218.5348	298.6645	1.00E-06	41.69405	272.3422	1.00E-06	20	0.182326
			40	11	95.98871	206.8298	6377.294	1.00E-06	51.75774	171.9698	1.00E-06	20	0.142439
		$ 14,332.16	1	12	182.9346	258.3584	247.5889	1.00E-06	41.65628	261.3008	1.00E-06	20	0.182629
			40	9	64.64786	138.0394	6361.352	1.00E-06	53.61987	162.0729	1.00E-06	20	0.138709
		$ 14,332.16	1	9	164.9576	213.4902	294.7095	1.00E-06	41.69573	262.8768	1.00E-06	20	0.182326
			40	8	97.02846	202.7234	6361.729	1.00E-06	66.89559	173.1741	1.00E-06	20	0.141749
		$ 13,435.46	1	9	164.9576	213.4902	294.7095	1.00E-06	41.69573	262.8768	1.00E-06	20	0.182326
			37	8	97.02846	202.7234	6361.729	1.00E-06	66.89559	173.1741	1.00E-06	20	0.141749
		$ 13,734.36	1	9	164.9576	213.4902	294.7095	1.00E-06	41.69573	262.8768	1.00E-06	20	0.182326
			38	8	97.02846	202.7234	6361.729	1.00E-06	66.89559	173.1741	1.00E-06	20	0.141749
		$ 7,421.63	1	13	166.0634	268.9902	283.7275	1.00E-06	37.79621	175.1596	1.00E-06	20	0.187575
			40	7	63.4922	131.0311	6634.142	1.00E-06	53.59991	195.5349	1.00E-06	20	0.108445
		$ 6,748.22	1	9	167.1768	265.8537	266.6752	1.00E-06	41.69032	253.9183	1.00E-06	20	0.182322
			34	8	98.2894	208.858	6362.776	1.00E-06	54.456	172.7944	1.00E-06	20	0.112571
		$ 14,332.16	1	7	187.2038	277.2427	262.5602	1.00E-06	41.68698	263.9617	1.00E-06	20	0.18233
			38	11	94.94089	116.6501	6365.978	1.00E-06	49.77393	155.4149	1.00E-06	20	0.113436
		$ 7,382.18	1	14	189.7403	223.6245	261.2626	1.00E-06	41.69741	215.315	1.00E-06	20	0.182649
			37	8	99.84924	202.715	6411.807	1.00E-06	55.80542	161.4165	1.00E-06	20	0.106406
		$ 14,332.16	1	9	168.5149	228.2915	295.3054	1.00E-06	41.63754	147.6531	1.00E-06	20	0.182406
			40	7	57.54147	214.8797	6533.245	1.00E-06	55.28407	164.9809	1.00E-06	20	0.10116
		$ 14,332.16	0	8	165.8653	212.4967	294.2688	1.00E-06	41.6413	263.3288	1.00E-06	20	0.18215
			42	8	96.57694	203.4553	6364.699	1.00E-06	66.7239	174.0435	1.00E-06	20	0.14184
		$ 14,332.16	0	12	234.1938	265.4982	261.0896	1.00E-06	37.54398	169.5609	1.00E-06	20	0.168898
			43	6	62.58531	130.7612	6588.302	1.00E-06	53.7984	239.493	1.00E-06	20	0.108151
		$ 14,332.16	1	9	290.1429	327.4012	223.4341	1.00E-06	34.24095	310.0528	1.00E-06	20	0.158198
1.9	11		38	7	33.69045	311.4138	2454.938	1.00E-06	39.07805	291.4083	1.00E-06	20	0.070043

Table E.2: Population and Scores for Fastest Solution, Middle East Target Deck

E.2 Ohio Target Deck

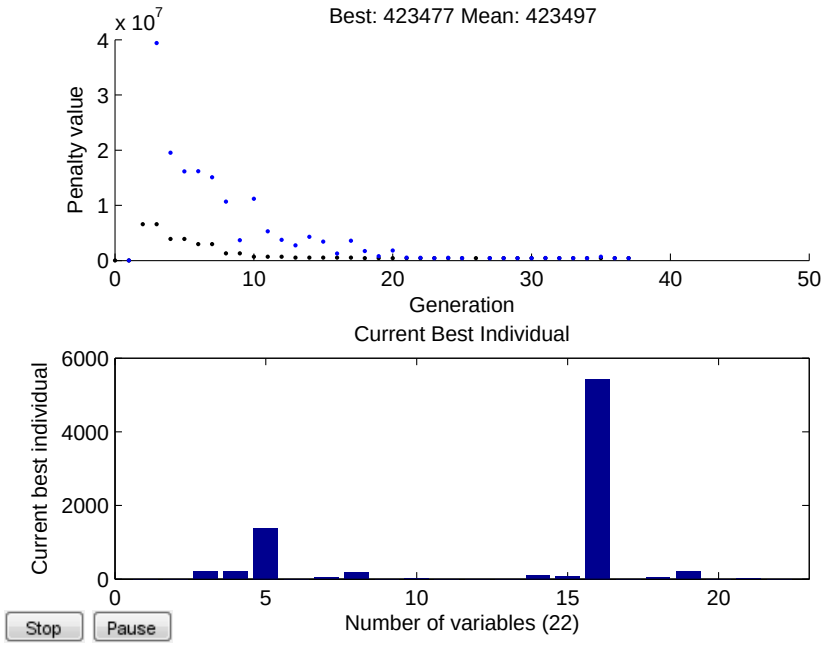


Figure E.3: Ohio Cheapest Solution. GA output plot for Ohio target deck with 60 minute constraint. Actual revisit time is 48.8 minutes.

OHIO - Cheapest													
		scores	population										
Revisit time	Generations	cost (millions)	nplanes	spp	raan inc	truanspr	alt	ecc	incl	raan	argp	M	diam (m)
		$ 104.62	1	1	212.0598	217.1345	1384.301	1.00E-06	41.24677	175.4806	1.00E-06	20	0.145602
			8	4	115.7564	76.43698	5435.335	1.00E-06	52.89459	212.1005	1.00E-06	20	0.052971
		$ 109.30	1	1	218.6208	217.2729	1384.314	1.00E-06	41.28224	175.4699	1.00E-06	20	0.145603
			8	4	115.7517	76.37873	5435.055	1.00E-06	52.90295	212.0393	1.00E-06	20	0.05297
		$ 113.78	1	1	210.603	217.2713	1384.113	1.00E-06	41.257	175.4801	1.00E-06	20	0.145603
			8	4	115.7563	76.40052	5437.242	1.00E-06	52.91018	212.0941	1.00E-06	20	0.052971
		$ 113.87	1	1	212.3619	217.2764	1384.172	1.00E-06	41.25639	175.4813	1.00E-06	20	0.145603
			8	4	115.7573	76.45706	5435.16	1.00E-06	52.89741	212.1023	1.00E-06	20	0.052971
		$ 113.87	1	1	210.1195	216.5924	1383.284	1.00E-06	41.24932	175.479	1.00E-06	20	0.145601
			8	4	115.7514	76.45202	5434.269	1.00E-06	52.87799	212.1343	1.00E-06	20	0.052972
		$ 113.87	1	1	212.3674	217.2797	1384.226	1.00E-06	41.25065	175.4813	1.00E-06	20	0.145603
			8	4	115.7573	76.4536	5434.629	1.00E-06	52.89812	212.0918	1.00E-06	20	0.052973
		$ 109.30	1	1	212.3156	217.4434	1384.318	1.00E-06	41.28117	175.4696	1.00E-06	20	0.145603
			8	4	115.7578	76.46877	5435.02	1.00E-06	52.89665	212.0399	1.00E-06	20	0.052972
		$ 113.87	1	1	212.0598	217.1345	1384.301	1.00E-06	41.24677	175.4806	1.00E-06	20	0.145602
			8	4	115.7564	76.43698	5435.335	1.00E-06	52.89459	212.1005	1.00E-06	20	0.052971
		$ 118.49	1	1	217.4999	216.593	1383.34	1.00E-06	41.19835	175.4673	1.00E-06	20	0.145602
			8	4	115.7519	76.44549	5436.995	1.00E-06	52.82026	212.0273	1.00E-06	20	0.052973
		$ 113.84	1	1	215.5695	217.2598	1381.152	1.00E-06	41.21487	175.4788	1.00E-06	20	0.145604
			8	4	115.7527	76.39723	5437.351	1.00E-06	52.81906	212.1032	1.00E-06	20	0.052971
		$ 113.87	1	1	210.3944	216.3497	1384.073	1.00E-06	41.21609	175.481	1.00E-06	20	0.145605
			8	4	115.7512	76.40046	5435.08	1.00E-06	52.8773	212.0448	1.00E-06	20	0.05297
		$ 118.52	1	1	210.8858	217.0824	1383.741	1.00E-06	41.22225	175.4693	1.00E-06	20	0.145601
			8	4	115.7522	76.38335	5435.465	1.00E-06	52.92744	212.0977	1.00E-06	20	0.05297
		$ 137.03	1	1	215.6852	216.3666	1384.084	1.00E-06	41.19967	175.4388	1.00E-06	20	0.145606
			8	4	115.7521	76.4646	5436.637	1.00E-06	52.82062	212.0386	1.00E-06	20	0.05297
		$ 104.62	1	1	218.8563	216.6934	1383.868	1.00E-06	41.24954	175.4737	1.00E-06	20	0.145607
			8	4	115.7559	76.38845	5436.85	1.00E-06	52.85582	212.0589	1.00E-06	20	0.052971
		$ 113.87	1	1	215.7329	216.9849	1384.532	1.00E-06	41.21396	175.4705	1.00E-06	20	0.145605
			8	4	115.7528	76.46851	5430.573	1.00E-06	52.81458	212.0446	1.00E-06	20	0.052968
		$ 123.19	1	1	218.17	216.3918	1381.521	1.00E-06	41.22958	175.4516	1.00E-06	20	0.145607
			8	4	115.7542	76.39444	5435.6	1.00E-06	52.81538	211.9781	1.00E-06	20	0.052969
		$ 127.76	1	1	218.3358	216.3716	1383.763	1.00E-06	41.21533	175.4683	1.00E-06	20	0.145605
			8	4	115.752	76.3886	5438.065	1.00E-06	52.82026	211.9932	1.00E-06	20	0.05297
		$ 137.03	0	0	212.0598	217.1345	1384.301	1.00E-06	41.24677	175.4806	1.00E-06	20	0.145602
			9	4	115.7564	76.43698	5435.335	1.00E-06	52.89459	212.1005	1.00E-06	20	0.052971
		$ 132.38	0	1	219.4634	219.1096	1395.84	1.00E-06	41.00145	178.986	1.00E-06	20	0.145584
			8	4	120.4081	74.94039	5329.579	1.00E-06	52.38521	207.9528	1.00E-06	20	0.052858
		$ 137.03	0	0	216.6857	216.4985	1382.416	1.00E-06	41.21146	175.6033	1.00E-06	20	0.145605
48.8	37		9	4	115.6652	76.60008	5437.709	1.00E-06	52.85622	212.2022	1.00E-06	20	0.052967

Table E.3: Population and Scores for Cheapest Solution, Ohio Target Deck

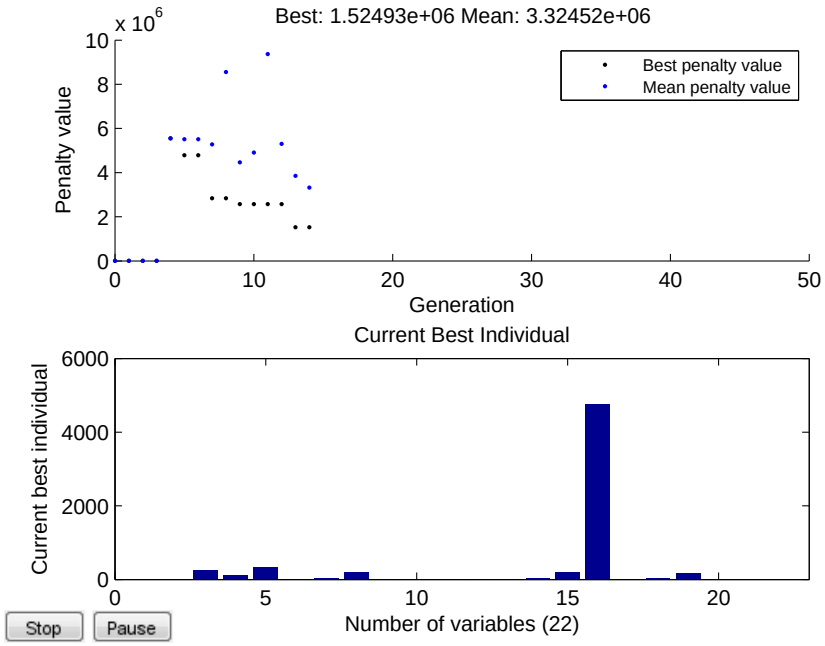


Figure E.4: Ohio Fastest Solution. GA output plot for Ohio target deck with 20 minute constraint. Actual revisit time is 12.9 minutes.

OHIO - Fastest													
		scores	population										
Revisit time	Generations	cost (millions)	nplanes	spp	raan inc	truanspr	alt	ecc	incl	raan	argp	M	diam (m)
		$ 1,524.93	1	1	260.167	114.9013	326.5041	1.00E-06	49.5723	202.3238	1.00E-06	20	0.177003
			9	9	46.31502	202.8849	4777.164	1.00E-06	43.96706	174.4656	1.00E-06	20	0.082307
		$ 1,745.70	2	1	259.3076	112.7852	327.5931	1.00E-06	49.74181	200.9602	1.00E-06	20	0.177283
			9	9	43.60854	204.6376	4758.029	1.00E-06	44.08729	172.8623	1.00E-06	20	0.08261
		$ 2,568.83	3	2	259.3059	112.7871	327.5953	1.00E-06	49.74112	200.9637	1.00E-06	20	0.177284
			9	9	43.60929	204.6349	4757.99	1.00E-06	44.08753	172.8652	1.00E-06	20	0.08261
		$ 2,834.88	4	2	258.4232	110.6201	328.7137	1.00E-06	49.91417	199.5696	1.00E-06	20	0.17553
			9	8	43.99156	203.273	4738.337	1.00E-06	44.21102	171.2248	1.00E-06	20	0.082922
		$ 2,835.19	4	2	258.4232	110.6609	328.7111	1.00E-06	49.91402	199.5689	1.00E-06	20	0.17553
			9	8	39.92838	201.6194	4738.337	1.00E-06	44.37288	171.2248	1.00E-06	20	0.082929
		$ 5,964.59	1	3	258.2495	94.15064	328.3121	1.00E-06	48.47961	200.3401	1.00E-06	20	0.347515
			8	8	40.30281	228.8348	4742.324	1.00E-06	44.94792	173.043	1.00E-06	20	0.082945
		$ 5,964.59	4	2	259.317	112.5379	327.5815	1.00E-06	49.91482	200.9943	1.00E-06	20	0.17742
			9	9	44.05199	186.2867	4764.501	1.00E-06	46.00052	170.9274	1.00E-06	20	0.082631
		$ 5,964.59	6	2	278.664	135.8224	284.1577	1.00E-06	44.47592	258.6536	1.00E-06	20	0.164261
			9	9	18.6491	211.6524	3197.872	1.00E-06	39.8777	117.0167	1.00E-06	20	0.109712
		$ 2,059.00	2	2	259.2874	116.9809	328.8615	1.00E-06	49.73482	199.2303	1.00E-06	20	0.17729
			9	8	43.73969	204.1781	4756.072	1.00E-06	44.03499	169.8194	1.00E-06	20	0.082923
		$ 5,964.59	4	2	258.4814	102.8432	328.6137	1.00E-06	49.91281	199.5708	1.00E-06	20	0.176687
			9	8	40.52805	187.9867	4727.335	1.00E-06	43.78316	171.4447	1.00E-06	20	0.082927
		$ 2,599.42	3	2	258.6278	103.161	328.6268	1.00E-06	49.74585	200.5515	1.00E-06	20	0.177601
			9	9	43.59117	207.1061	4743.44	1.00E-06	44.02601	172.966	1.00E-06	20	0.083163
		$ 2,548.43	3	2	258.4635	108.2385	327.5173	1.00E-06	49.73772	201.0173	1.00E-06	20	0.176519
			9	9	43.56295	204.2662	4735.169	1.00E-06	44.06482	171.127	1.00E-06	20	0.082516
		$ 2,359.16	3	2	258.4657	134.3829	328.5379	1.00E-06	48.82297	199.5316	1.00E-06	20	0.171976
			9	8	41.01477	192.2205	4729.613	1.00E-06	45.57302	171.5442	1.00E-06	20	0.082931
		$ 2,153.31	1	4	259.4076	115.232	328.1444	1.00E-06	49.84892	201.1129	1.00E-06	20	0.177083
			9	9	39.61691	206.7088	4761.288	1.00E-06	43.99374	169.2808	1.00E-06	20	0.082592
		$ 1,941.32	3	1	260.1641	114.7636	326.6113	1.00E-06	49.58026	201.0015	1.00E-06	20	0.177364
			9	9	46.37048	202.9878	4776.265	1.00E-06	44.07507	172.6132	1.00E-06	20	0.082345
		$ 2,155.62	4	1	260.2549	114.791	326.4908	1.00E-06	49.73616	200.896	1.00E-06	20	0.177221
			9	9	46.39364	202.9341	4764.058	1.00E-06	43.96357	173.5453	1.00E-06	20	0.082595
		$ 1,527.54	1	1	260.1799	114.918	327.8772	1.00E-06	49.57643	201.6382	1.00E-06	20	0.177333
			9	9	44.13146	203.1017	4759.441	1.00E-06	43.93478	172.7212	1.00E-06	20	0.082309
		$ 1,849.51	3	1	258.3448	254.706	328.723	1.00E-06	49.91765	199.5506	1.00E-06	20	0.176997
			9	8	38.94225	204.1898	4735.608	1.00E-06	44.28629	169.8322	1.00E-06	20	0.082927
		$ 5,964.59	4	2	254.3491	102.7987	326.5488	1.00E-06	49.60451	202.1019	1.00E-06	20	0.178739
			10	8	40.41482	188.5519	4765.176	1.00E-06	45.73672	174.2112	1.00E-06	20	0.082379
		$ 5,964.59	4	1	258.8546	103.8824	328.1076	1.00E-06	49.22873	198.8392	1.00E-06	20	0.178258
12.89	14		9	10	37.70781	229.1732	4742.399	1.00E-06	46.15581	170.2089	1.00E-06	20	0.083015

Table E.4: Population and Scores for Fastest Solution, Ohio Target Deck

Bibliography

1. Taverney, T., "Resilient, Disaggregated, and Mixed Constellations," *Space Review*, 29 August 2011, pp. 15 September 2013, Accessed: 2013-09-15.

2. Burch, R., "OPS: Disaggregation + Diversification of U.S. MILSATCOM," `https://www.milsatmagazine.com/story.php?number=510061234`, April 2012, Accessed: 2013-10-07.

3. "Defense Acquisition Guidebook," 16 Sep 2013, pp. 3.7.2.4, Accessed: 2013-01-02.

4. Wertz, J. R., Everett, D. F., and Puschell, J. J., *Space Mission Engineering: The New SMAD*, Microcosm Press, Hawthorne, CA, 2011.

5. Brown, O. and Eremenko, P., "Fractionated Space Architectures: A Vision for Responsive Space," Tech. rep., Defense Advanced Research Projects Agency (DARPA), Jan 2008.

6. Co, T. C., "Operationally Responsive Spacecraft Using Electric Propulsion," 23 July 2012.

7. Co, T. C., Zagaris, C., and Black, J. T., "Responsive Satellites through Ground Track Manipulation using Existing Technology," *Journal of Spacecraft and Rockets*, Vol. 50, No. 1, Jan. - Feb. 2013, pp. 206–216.

8. Co, T. C. and Black, J. T., "Responsiveness in Low Orbits using Electric Propulsion," *Journal of Spacecraft and Rockets*, Accepted for publication Apr. 2013.

9. Pawlikowski, E., Loverro, D., and Cristler, T., "Space: Disruptive Challenges, New Opportunities, and New Strategies," *Strategic Studies Quarterly*, Spring 2012, pp. 27.

10. Brown, O. and Eremenko, P., "Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft," *AIAA*, 2008.

11. AFSPC, "Resiliency and Disaggregated Space Architectures (White paper)," 2013.

12. Anhalt, D., "National Security Space Strategic Perspectives on Resilience in a Cost Constrained Environment," *AIAA SPACE 2012 Conference*, 2012.

13. Brown, O., Eremenko, P., and Roberts, C., "Cost-Benefit Analysis of a Notional Fractionated SATCOM Architecture," *24th AIAA International Communications Satellite Systems Conference (ICSSC)*, 2006.

14. Tozer, J. L., "Untouchable: Fighting Forward in the Space, Cyber Domains," `http://science.dodlive.mil/2013/09/20/untouchable-fighting-forward-in-the-space-cyber-domains/`, September 20, 2013, Accessed: 2013-10-07.

15. Gruss, M., "Shelton: Sequestration Could Break Military Space Program," http://www.spacenews.com/article/military-space/37270shelton-sequestration-could-break-military-space-program, September 17, 2013, Accessed: 2013-10-07.

16. Garamone, J., "Shelton Discusses Importance of Space Defense," http://www.defense.gov/news/newsarticle.aspx?id=121443, Jan 7, 2014, Accessed: 2013-01-08.

17. Jilla, C. D., *A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems*, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2002, Accessed: 2013-10-02.

18. Horst, J. V. D., Noble, J., and Tatnall, A., "Robustness of market-based task allocation in a distributed satellite system," *Proceedings of the 10th European conference on Advances in artificial life: Darwin meets von Neumann - Volume Part II*, ECAL'09, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 334–341, Accessed: 2013-10-07.

19. Ferster, W., "DARPA Cancels Formation-Flying Satellite Demo," http://www.spacenews.com/article/military-space/35375darpa-cancels-formation-flying-satellite-demo, 17 May 2013, Accessed: 2013-09-20.

20. Sellers, J. J., Astore, W. J., Giffen, R. B., and Larson, W. J., *Understanding Space*, McGraw-Hill, New York, NY, 2nd ed.

21. Mathieu, C. and Weigel, A. L., "Assessing the Flexibility Provided by Fractionated Spacecraft," Tech. rep., American Institute of Aeronautics and Astronautics, Inc., 30 August - 1 September 2005.

22. DARPA, "System F6," http://www.darpa.mil/our_work/tto/programs/system_f6.aspx, Accessed: 2013-09-20.

23. Larson, W. J., Kirkpatrick, D., Sellers, J. J., Thomas, L. D., and Verma, D., *Applied Space Systems Engineering*, McGraw-Hill, New York, NY, 2009.

24. STK10, "STK Programming Interface Help," .

25. Delgado, L. M., "Benefits Weighed of Disaggregation for Military Space Systems," http://www.spacepolicyonline.com/news/benefits-weighed-of-disaggregation-for-military-space-systems, 30-Jan-2013, Accessed: 2013-10-07.

26. Gruss, M., "Senate Bill Would Kill DARPA's SeeMe Project," http://www.spacenews.com/article/military-space/36860senate-bill-would-kill-darpas-seeme-project, Aug 20, 2013, Accessed: 2013-11-18.

27. London, J. R., Marley, A. B., and Weeks, D. J., "PRIME: Nanosat Demos For The Tactical Land Warfighter," http://www.milsatmagazine.com/story.php?number=1752663203, March 2012, Accessed: 2013-11-18.

28. USASMDC/ARSTRAT, "NanoEye Fact Sheet," http://www.smdc.army.mil/FactSheets/NanoEye.pdf, Accessed: 2014-02-07.

29. Ingraham, S., "Dynamic Constellation Tasking and Management," 2013.

30. USASMDC/ARSTRAT, "Kestrel Eye Fact Sheet," http://www.smdc.army.mil/FactSheets/KestrelEyeTC0112.pdf, Accessed: 2014-02-07.

31. "Army Small Business Innovation Research (SBIR) 13.2 Topic Descriptions," 2013.

32. Taylor, T., "Low-Cost Tactical Space Capabilities," http://smdsymposium.org/wp-content/uploads/2013/09/Travis-Taylor-Presentation.pdf, 14 August 2013 2013, Accessed: 2014-02-07.

33. Ferster, W., "For a Revamped GPS, Biggest Savings Carry Biggest Risks," http://www.spacenews.com/article/military-space/37315for-a-revamped-gps-biggest-savings-carry-biggest-risks, Sep. 23, 2013, Accessed: 2013-10-07.

34. Battistelli, E., Butera, F., Gonzalo, J., and Jimenez-Tunon, L., "FUEGOSAT: The Space Payload for Fire Observation," *Galileo Avionica*, 2005, pp. 134.

35. Emery, J. D., Oberg, E. M., Robertson, K. A., Sanchez, D. H., and Smuck, D. B., "The Utility and Logistics Impact of Small-Satellite Constellations in Matched Inclination Orbits," March 2005.

36. Mathieu, C. and Weigel, A. L., *Assessing the Flexibility Provided by an On-orbit Infrastructure of Fractionated Spacecraft*, American Institute of Aeronautics and Astronautics, 17 Oct 2005, 23; M1: 0; doi:10.2514/6.IAC-05-D3.3.01; M3: doi:10.2514/6.IAC-05-D3.3.01.

37. USDOT and FAA, "Commercial Space Transportation Quarterly Launch Report," Tech. rep., Associate Administrator for Commercial Space Transportation, 4th Quarter 2002, Used for Launch Insurance figure.

38. Liu, X., Chen, Y., Chen, Y., and He, R., "Optimization of earth observation satellite system based on parallel systems and computational experiments," *Journal of Control Theory and Applications*, Vol. 11, No. 2, 2013, pp. 200–206.

39. deSelding, P. B., "Debris-control Report Card Cites Improvement by Geo Sat Owners," http://www.spacenews.com/article/satellite-telecom/37861debris-control-report-card-cites-improvement-by-geo-sat-owners, Oct. 25, 2013.

40. Mathieu, C. and Weigel, A. L., "Assessing the Fractionated Spacecraft Concept," *Space*, 19-21 September 2006.

41. Arora, J. S., *Introduction to Optimum Design*, Academic Press, New York, NY, 3rd ed., 2012.

42. Patalia, T. P. and Kulkarni, G. R., "Behavioral analysis of genetic algorithm for function optimization," *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, 2010, pp. 1–5.

43. Hu, X., "PSO Tutorial," `http://www.swarmintelligence.org/tutorials.php`, 2006, Accessed: 2014-2-28.

44. Reyes-Sierra, M. and Coello-Coello, C. A., "Multi-Objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, 2006, pp. 287–308.

45. Sales, J., "Trajectory Optimization for Spacecraft Collision Avoidance," 2013.

46. Irvine, J. M., "National Imagery Interpretability Rating Scales (NIIRS): Overview and Methodology," *Airborne Reconnaissance XXI*, Vol. 3128, November 21, 1997, p. 93.

47. NASA, "Landsat Science Data Users Handbook," March 11, 2011, Accessed: 2014-3-10.

48. Werner, D., "Earth Science and Climate Monitoring — Scientists Study How Satellites Could Improve Volcano Forecasts," `http://www.spacenews.com/article/features/37249earth-science-and-climate-monitoring-scientists-study-how-satellites-could`, 16 Sep 2013, Accessed: 2013-09-20.

49. Company, R., "Joint Polar Satellite System Common Ground System," `http://www.raytheon.com/capabilities/products/jpss/index.html`, 2013, Accessed: 2013-10-15.

50. Company, R., "Joint Polar Satellite System Budget Recommendations," `http://www.raytheon.com/capabilities/products/jpss/budget/index.html`, 2013, Accessed: 2013-10-15.

51. Kramer, H. J., "Ikonos-2," `http://www.eoportal.org/directory/pres_Ikonos2.html`, 24 Jan 2008, Accessed: 2013-01-07.

52. Sales, J., "STK Library," 2013.

53. Matlab, "Matlab help," .

54. Auelmann, R. R., "Image Quality Metrics," *Richard R. Auelmann Techical Archive*, 5 Nov 2012.

55. Gruss, M., 7 Mar 2014.

56. Thurman, S. T. and Fienup, J. R., "Analysis of the General Image Quality Equation," *Proc. SPIE*, Vol. 6978, March 25, 2008, Accessed: 2013-12-09.

Vita

Evelyn Abbate graduated from Egg Harbor Township High School in New Jersey in 2008, and entered undergraduate studies at the United States Air Force Academy. While a cadet, she was qualified in all crew positions for FalconSat-3, and also operated communications and antenna systems for FalconSat-5. In her free time, she played violin with the Cadet Orchestra and served as cadet in charge of the Aikido club. As a senior with a minor in Japanese, she received the opportunity to spend a semester at the Japanese National Defense Academy. She graduated with a commission to the United States Air Force and a Bachelor of Science degree in Astronautical Engineering in 2012. In 2014, she graduated from the Masters program at the Air Force Institute of Technology, and has been assigned to the National Reconnaissance Office (NRO) in Chantilly, VA.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
27-03-2014	Master's Thesis	Oct 2012 - Mar 2014

4. TITLE AND SUBTITLE

Disaggregated Imaging Spacecraft Constellation Optimization with a Genetic Algorithm

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Abbate, Evelyn A., 2Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-14-M-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A. Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S.
Government and is not subject to copyright protection in the United States.

14. ABSTRACT

This research is an extension of work by Major Robert Thompson, who uses a genetic algorithm to optimize certain parameters of a disaggregated constellation for most cost-effective coverage. This work looks at imaging sensor coverage of a specific target deck assumed to exist in the Middle East. Parameters varied in this optimization affect Walker constellation characteristics, orbital elements, and sensor size. Walker parameter variables are number of planes, number of satellites per plane, true anomaly spread, and RAAN increment. All classical orbital elements are variable, although a circular, low-Earth orbit is assumed. Sensor size is varied dependent upon sensor diameter. These parameters are applied to constellations of small satellites and large satellites. The Unmanned Spacecraft Cost Model (USCM) and the Small Spacecraft Cost Model (SSCM) are used to roughly determine the cost of each proposed mission. The sensor effectiveness is determined by the General Imaging Quality Equation (GIQE).

15. SUBJECT TERMS

Disaggregation, optimization, Earth observation, space architecture, responsive space

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Jonathan T. Black AFIT/ENY
UU	UU	UU	UU	156	19b. TELEPHONE NUMBER *(Include area code)* (937) 785-3636 x4578 Jonathan.Black@afit.edu